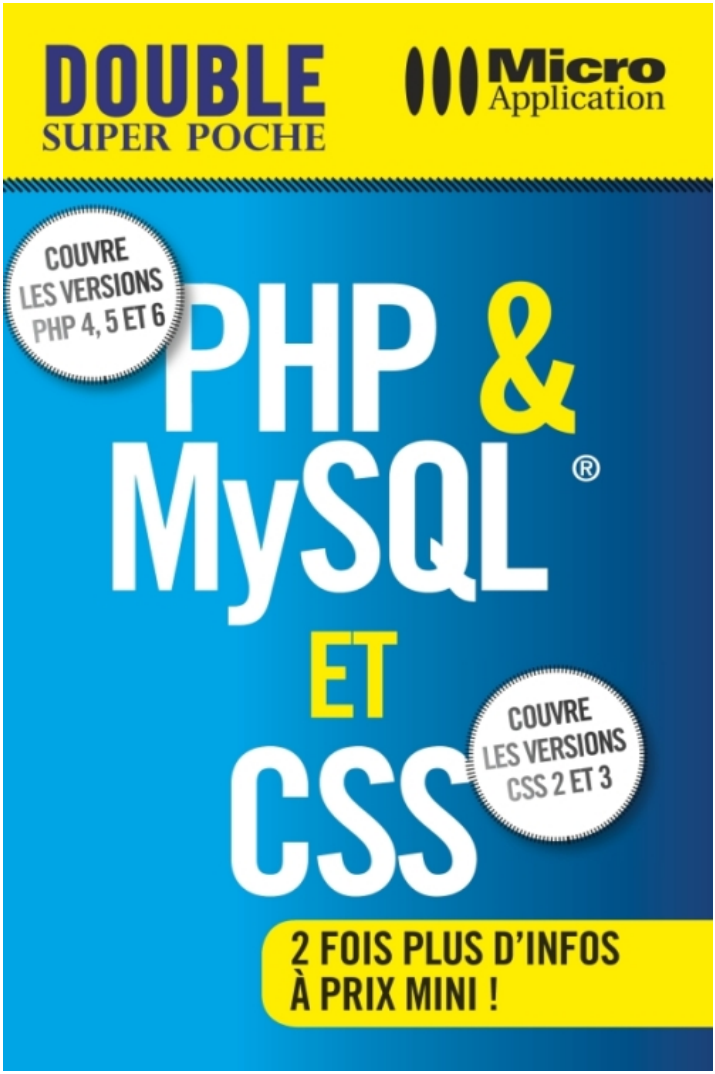
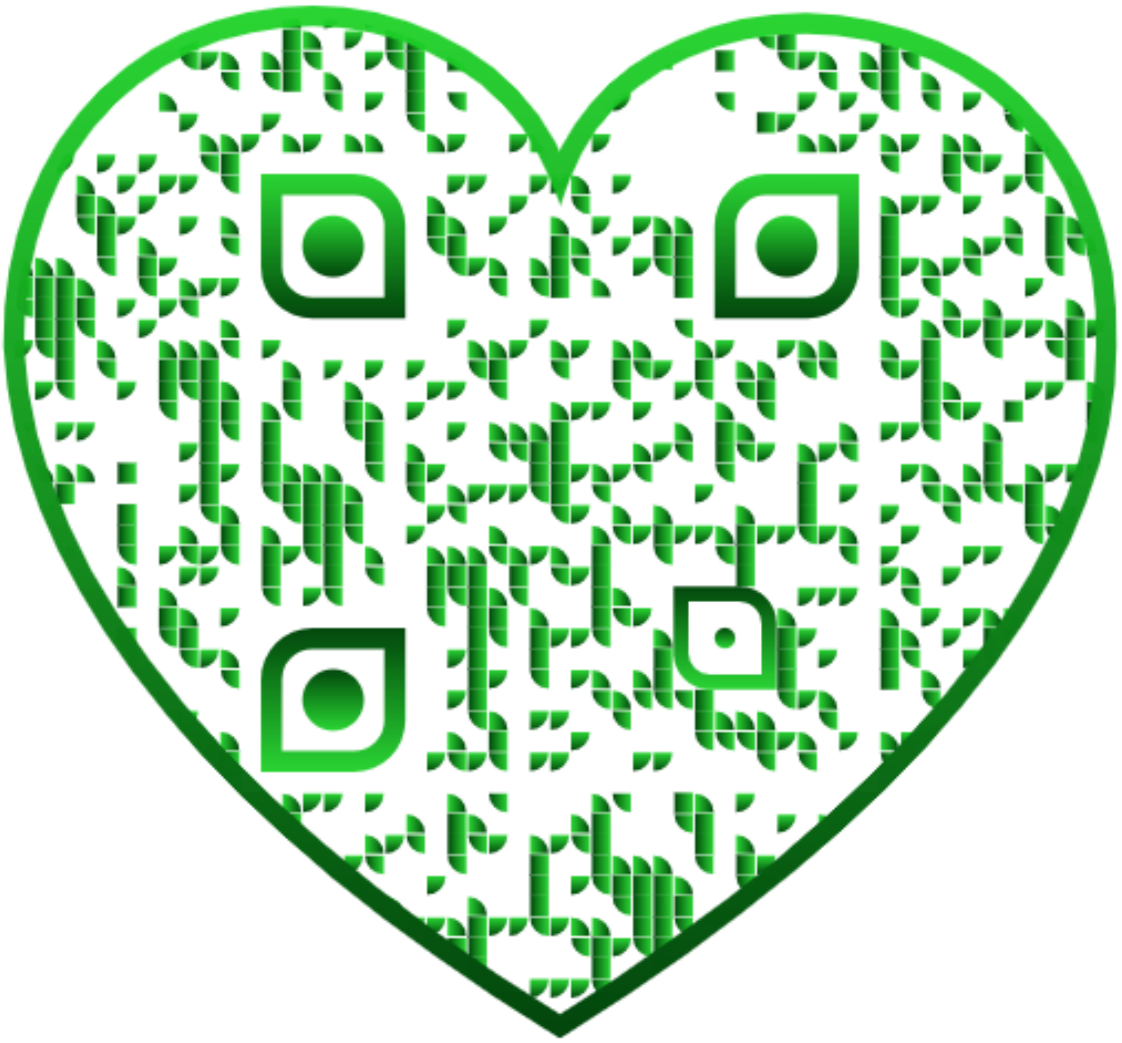

DOUBLE
SUPER POCHE
Micro Application
COUVRE LES VERSIONS PHP 4, 5 ET 6
PHP & MySQL®
ET
CSS
COUVRE LES VERSIONS CSS 2 ET 3
2 FOIS PLUS D'INFOS
À PRIX MINI !



Copyright

© 2009 Micro Application
20-22, rue des Petits-Hôtels
75010 PARIS

1re Edition - Mars 2009

Auteur

Jean CARFANTAN

Toute représentation ou reproduction, intégrale ou partielle, faite sans le consentement de MICRO APPLICATION est illicite (article L122-4 du code de la propriété intellectuelle).
Cette représentation ou reproduction illicite, par quelque procédé que ce soit, constituerait une contrefaçon sanctionnée par les articles L335-2 et suivants du code de la propriété intellectuelle.
Le code de la propriété intellectuelle n'autorise aux termes de l'article L122-5 que les reproductions strictement destinées à l'usage privé et non destinées à l'utilisation collective d'une part, et d'autre part, que les analyses et courtes citations dans un but d'exemple et d'illustration.

Avertissement aux utilisateurs

Les informations contenues dans cet ouvrage sont données à titre indicatif et n'ont aucun caractère exhaustif voire certain. A titre d'exemple non limitatif, cet ouvrage peut vous proposer une ou plusieurs adresses de sites Web qui ne seront plus d'actualité ou dont le contenu aura changé au moment où vous en prendrez connaissance.
Aussi, ces informations ne sauraient engager la responsabilité de l'Editeur. La société MICRO APPLICATION ne pourra être tenue responsable de toute omission, erreur ou lacune qui aurait pu se glisser dans ce produit ainsi que des conséquences, quelles qu'elles soient, qui résulteraient des informations et indications fournies ainsi que de leur utilisation.
Tous les produits cités dans cet ouvrage sont protégés, et les marques déposées par leurs titulaires de droits respectifs. Cet ouvrage n'est ni édité, ni produit par le(s) propriétaire(s) de(s) programme(s) sur le(s)quel(s) il porte et les marques ne sont utilisées qu'à seule fin de désignation des produits en tant que noms de ces derniers.

ISBN : 978-2-300-017599

Couverture réalisée par Emilie Greenberg

MICRO APPLICATION
20-22, rue des Petits-Hôtels
75010 PARIS
Tél. : 01 53 34 20 20
Fax : 01 53 24 20 00
http://www.microapp.com

Support technique
Également disponible sur
www.microapp.com

Retrouvez des informations sur cet ouvrage !

Rendez-vous sur le site Internet de Micro Application **www.microapp.com**. Dans le module de recherche, sur la page d'accueil du site, entrez la référence à 4 chiffres indiquée sur le présent livre. Vous accédez directement à sa fiche produit.

Avant-propos

Cette collection fournit des connaissances essentielles sur un sujet donné sans jamais s'éloigner de leur application pratique. Les volumes de la collection sont basés sur une structure identique :

- Les puces introduisent une énumération ou des solutions alternatives.

1. La numération accompagne chaque étape d'une technique.

Il s'agit d'informations supplémentaires relatives au sujet traité.

Met l'accent sur un point important, souvent d'ordre technique, qu'il ne faut négliger à aucun prix.

Propose conseils et trucs pratiques.

Conventions typographiques

Afin de faciliter la compréhension des techniques décrites, nous avons adopté les conventions typographiques suivantes :

- **Gras** : menu, commande, boîte de dialogue, bouton, onglet.
- *Italique* : zone de texte, liste déroulante, case à cocher, bouton radio.
- `Police bâton` : touche, instruction, listing, texte à saisir.
- ✂ : indique un retour ligne volontaire dû aux contraintes de la mise en page.

PHP & MySQL

1 Introduction 17

1.1. Naissance de ce langage 18
1.2. Evolution du langage 18
1.3. Intérêt de ce langage 19
La liberté 19
La portabilité 20
La facilité 21
Un peu d'histoire 21
1.4. Fonctionnement 22
1.5. Des chiffres. 23

2 Ma première page PHP 25

2.1. HTML 26
2.2. Le cheminement d'un apprenti webmestre 27
2.3. Quelques notions de HTML 28
Les images 32
Les liens 33
Les tableaux 35
Les formulaires 38
Les cadres ou frames 40
2.4. Les outils 43
Les logiciels d'édition de code PHP 43
EasyPHP 45
Autres logiciels 46
L'hébergement 47
2.5. Ma page 48
Les commentaires 51
PHPinfo() 51

3 Les clefs du PHP 53

3.1. Variables scalaires 54
Le nom de variable 54
Le type des variables 55
Les guillemets 57
Les constantes 60
Les tableaux 63
3.2. Les opérateurs 67
Les opérateurs arithmétiques 67
Les opérateurs d'affectation 67
Les opérateurs de bits 68
Les opérateurs de comparaison 68
Les opérateurs de contrôle d'erreur 70
Les opérateurs unaires 71
Les opérateurs logiques 72
Priorité des opérateurs 72

Les opérateurs de chaîne de caractères ... 73
3.3. Les structures de contrôle ... 74
Les tests ... 74
3.4. Les boucles ... 77
La boucle while ... 78
La boucle do while ... 79
La boucle for ... 80
Foreach ... 82
Break et continue ... 84
3.5. Les fonctions de PHP ... 85
Chaînes de caractères ... 86
Variables ... 91
Date et heure ... 91
Mathématiques ... 95
3.6. Les fonctions ... 97
Le rôle de la fonction ... 97
La portée d'une variable dans une fonction ... 101
Les fonctions imbriquées ... 101
Les fonctions récursives ... 102
Afficher vos date et heure locales ... 103
3.7. Récapitulatif pour votre application ... 105

4 MySQL en action ... 107

4.1. Le fichier face à la base de données ... 108
4.2. Conception de l'application du club ... 108
4.3. Création d'une table ... 109
4.4. Modification d'une table ... 113
4.5. Les clés étrangères ... 114
4.6. Les jointures de tables ... 115
4.7. Les types de données MySQL ... 116
4.8. La sauvegarde de la base ... 118
4.9. Les formulaires ... 119
L'inscription au club ... 119
4.10. Les fonctions du langage PHP pour MySQL ... 133
4.11. La vérification des formulaires ... 133
La vérification avec Javascript ... 134
Le code PHP ... 141
Variables prédéfinies liées au chemin ou à l'URL ... 145
4.12. Tout sur un seul fichier : oubli.php ... 146
La réflexion ... 146
La plate-forme commune ... 147
Les variables comme marqueurs ... 147
4.13. Récapitulation de la syntaxe des requêtes ... 151
4.14. Les procédures stockées ... 156
4.15. SQLite ... 156

5 Le PHP et l'approche objet 159

5.1. Qu'est-ce que l'objet ? 161
5.2. L'héritage 163
5.3. Surcharger une méthode 165
5.4. Le polymorphisme 166
5.5. Passer un argument par référence 167
5.6. PHP 5 et l'objet 168
Le pointeur 168
Les privilèges 168
Le constructeur et le destructeur 168

6 L'internaute est d'abord une personne 171

6.1. L'authentification 172
Le processus 175
La première authentification 176
La vérification de l'identité de l'internaute 178
Le cookie 179
La conservation des données personnelles 183
6.2. Le tableau de bord du membre 184
Modifier ou supprimer un profil 185
6.3. Le tableau de bord d'administration du site 190
L'accès restreint d'un répertoire avec .htaccess 191
Le tableau de bord 192
6.4. Des aménagements dans l'application 198
6.5. En-têtes HTTP et variables globales 201
En-têtes HTTP 201
Variables PHP 204
Variable d'environnement 205
6.6. Récapitulation 207

7 Des algorithmes, des outils et des fonctions 209

7.1. Tableaux 210
Types d'extraction 212
Le pointeur 213
Le tri 213
7.2. Calendriers, dates et autres agendas 218
Durée après une date ou avant une date 218
Affichage d'un calendrier 219
Un agenda à jour 223
7.3. Un miniforum sur chaque page 236
7.4. Sondages et concours 239
Un sondage 240
Un concours 247
Les fonctions concernant le système de fichiers 249
7.5. Compteurs et statistiques 254

La réflexion 254
Créer le compteur quotidien 254
La page d'affichage des statistiques de page 256
7.6. Le chariot électronique 258
La promotion 259
Les prix 260
Les tables à créer 261
L'action "acheter un produit" 261
L'action "voir le contenu du chariot" 262
L'action "commander les articles" 263
7.7. Un magazine 265
La réflexion 266
Le formulaire pour entrer les images 267
7.8. Une galerie photo 270
Les données EXIF 272

8 Moteurs de recherche et expressions rationnelles . 275

8.1. Un annuaire de liens 276
Analyse du besoin 276
Réflexions 277
Élaboration des applications 279
Moteur de recherche 295
8.2. Les expressions rationnelles 301
La méthode 301
POSIX 302
Les fonctions compatibles Perl 307
8.3. Conclusion 319

9 Parsez-vous XML ? 321

9.1. Un peu d'histoire... 322
9.2. Le fichier XML 323
L'espace de nommage 323
Les éléments et les attributs 323
9.3. La DTD 326
Les différents types de déclarations 326
9.4. Le XSLT 328
L'espace de nommage 328
La feuille de style 329
Les modèles 330
9.5. Le fichier PHP 332
9.6. Magasin électronique en XML 332
La conception de la boutique 333
Création du fichier XML 333
Création d'un nouveau fichier CSV à partir d'un fichier XSLT 335
Le fichier XSLT de la boutique 336
Le fichier PHP de la boutique 339

9.7. SimpleXML 340
9.8. Conclusion 342

10 Les frameworks ou cadriciels PHP 345

10.1. Une méthodologie de travail 347
10.2. L'architecture Modèle/Vue/Contrôleur 348
Modèle 348
Vue 348
Contrôleur 349
10.3. Xataface, donnez un visage humain à votre base de données .. 349
L'architecture 350
Les tables 352
Application Déléguée dans le répertoire conf 354
Conclusion 354
10.4. CakePHP, c'est du gâteau 355
10.5. JOOMLA, le meilleur gestionnaire de contenu 357
Drupal contre Joomla 360
10.6. Conclusion 361

11 Mettre de l'ordre 363

11.1. Avant 365
11.2. Après 366
11.3. Le débogage 366
11.4. Sachez anticiper 369

12 La traversée de l'écran 371

12.1. La communication sur le Web 373
L'humour 374
Le soutien 374
Les émoticônes 375
La nétiquette 375
La mise à jour régulière 376
La navigation 376
Le référencement 377
12.2. Conclusion 377

13 PHP 6 en ligne de mire 379

13.1. Unicode 380
13.2. Le paramètre Register Globals n'est plus disponible 381
13.3. Les Magic Quotes sont retirées 382
13.4. Le Safe Mode n'existe plus 382
13.5. Utiliser Freetype 2 et GD 2 383
13.6. Plus de HTTP_*_VARS 383

13.7. Les extensions in et out 383
13.8. Les ajouts 383

14 Annexe 387

14.1. PHP 388
PHP Hypertext Preprocessor 388
PHP Builder, the Resource for PHP developers 389
PHP Index, la Passerelle française des technologies PHP 389
Nexen 390
PHP Facile 391
ASP-PHP.NET 391
PHP France 392
PHP Heaven 393
AFUP 393
14.2. MySQL 395
MySQL 395
Nexen 396
14.3. Autres 396
Apache 396
Easyphp 397
14.4. Outils 398
Éditeurs 398
FTP 400
Personnels 401
14.5. Applications en PHP 403
14.6. Les portails 405
Typo3 405
Spip 405
14.7. Hébergement 405
Gratuit 405
Payant 406
Noms de domaine 407
14.8. Référencement 407
Référencement 2000 407
14.9. HTML et Javascript 407
All HTML 407
Le Journal du net 408
Comment ça marche 408
14.10. Les groupes de news 408
Pour écrire à l'auteur de ce livre 408

15 Index 409

Préface

Ce livre s'adresse à tous les publics. Que vous soyez débutant, webmestre ou programmeur, vous tirerez profit de cet ouvrage en vous amusant. Si vous avez l'intention de créer un site Internet mais que vous vous sentiez démuni, incapable, dépassé, PHP & MySQL en ligne constitue le début idéal de votre aventure. Les pages qui vont suivre sont issues de ma pratique quotidienne d'analyste-programmeur et de webmestre. J'ai veillé à parsemer ce livre d'astuces et d'exemples complets avec des explications claires et une évolution progressive.

Pour cette deuxième édition, nous abordons PHP 5, qui entame une vraie révolution dans la manière d'appréhender le code et la grammaire. PHP 5 garde cependant une compatibilité ascendante qui garde opérationnelles des fonctions obsolètes. Les points saillants de cette révolution sont le renforcement de la programmation objet, l'arrivée d'une nouvelle base de données *SQLite* et un vrai espace pour le XML.

Le langage PHP est un vrai continent, composé, pour la version 4, de plus de 2 700 fonctions et de 80 extensions. Au cours de notre exploration, vous apprendrez des méthodes et maîtriserez progressivement les outils dans le but de créer un site interactif. Vous étudierez donc les principales fonctions et notions du langage PHP. À travers des exemples, vous verrez comment créer facilement vos applications. En douceur et sans vous encombrer de notions superflues, vous assemblerez ces applications simples pour réaliser une application "complexe".

Ce livre est fondé sur mon expérience. En 1994, j'étais vendeur de vidéo numérique dans une grande surface informatique (Surcouf) ; c'est là que j'ai découvert l'Internet. À cette époque, il existait peu de sites en français. J'ai d'ailleurs réalisé mon premier site en anglais. Après avoir enrichi ce site et arpenté longtemps l'Internet, j'ai voulu aller plus loin. C'est là que j'ai découvert le langage Perl, sous la forme d'un script capable de traiter des données de formulaire envoyées par e-mail. Par simple curiosité, j'ai acheté un livre, puis deux… et j'ai été happé par la programmation. Seul, j'ai donc réalisé des algorithmes de plus en plus complexes et acquis des méthodes.

Après cinq ans de pratique, je suis entré en 1999 à l'IUT de Marne-la-Vallée pour étudier d'autres méthodes de programmation et aborder Java et le langage objet. À la sortie de l'université, mon diplôme en poche, j'ai intégré Nexus Développement, où j'ai pratiqué le langage PHP à temps complet pour le Web et expérimenté les bonnes (et mauvaises) méthodes de programmation.

Pour restituer un savoir, il est nécessaire de l'expérimenter, c'est pourquoi j'ai emprunté pour vous des chemins que je n'avais jamais foulés auparavant. De ce fait, ce livre est utile pour des programmeurs chevronnés.

Au long de cet ouvrage, j'ai eu à cœur de rester accessible à tous. Quels que soient votre âge et vos connaissances, vous pourrez programmer. La complexité réside non pas dans le langage PHP — l'outil est simple —, mais dans la conception de votre programme, dans son architecture et sa richesse.

Un langage est fait pour la communication hommes/hommes, hommes/machines et machines/machines. La plus grande compétence d'un analyste-programmeur reste sa capacité à concevoir, à communiquer et à adapter son outil. L'outil est au service d'une fonction. Les langages deviennent trop souvent des chapelles de spécialistes, de savantes constructions mentales qui perdent de vue leur fonction et brillent uniquement dans les yeux des experts. Périodiquement, des langages tombent dans l'oubli parce qu'ils ont oublié la vie. L'informatique, comme le droit ou la médecine, peut malheureusement devenir une affaire d'experts en jargon. Ainsi, certains spécialistes sont à même de manipuler un client pour le faire payer des prestations interminables. Le langage PHP est différent.

J'aimerais que ce langage soit enseigné à l'école car les élèves peuvent programmer facilement et obtenir des résultats immédiats, ce qui les encourage à approfondir. Tout langage de programmation enseigne la rigueur et la structure. La programmation informatique ouvre sur les mathématiques, bien sûr, par la logique, mais aussi sur la grammaire et la rédaction, par son souci de précision et la nécessité de communiquer convenablement. Il est indispensable d'être compris par l'internaute sur un site mais également par le collègue qui reprendra le programme derrière vous.

Je dédie ce livre à mon père, Maurice, qui a toujours encouragé mon goût d'apprendre, et à mon filleul cambodgien, Touch Ra, qui un jour, je l'espère, pourra lire cet ouvrage et s'amuser avec du code informatique, s'il le souhaite.

1

Introduction

PHP. Encore un sigle comme l'informatique les affectionne. On fait souvent des plaisanteries sur leur signification. Pourtant, les inventeurs de langage, ces forgerons de la syntaxe, possèdent aussi de l'humour, c'est pourquoi fréquemment, surtout dans la communauté du Libre (ou Open Source), le nom du langage est fondé sur une plaisanterie pour initiés (private joke). Ainsi, comme Java signifie café en argot américain, il faut croire que la cafetière a fourni le carburant aux créateurs de ce langage. PHP est d'abord le sigle de Personal Home Page. Rasmus Lerdorf, son concepteur, s'est souvenu des fonctions récursives et l'a baptisé (PHP Hypertext processor). Si vous développez le sigle, vous générez une boucle infinie, crainte des programmeurs. PHP signifie PHP Hypertext Processor Hypertext Processor Hypertext Processor… C'est une fonction "Vache qui rit", suivant la logique de cette boîte de fromage célèbre qui affiche une vache avec des boîtes comme boucles d'oreille affichant une vache avec des boîtes… Justement, les boucles sont les éléments qui caractérisent un langage de programmation.

Reportez-vous au chapitre intitulé **Les clefs du PHP** pour obtenir plus de détails sur la fonction récursive et les boucles.

1.1 Naissance de ce langage

Le langage PHP est né en 1994 d'un programme en langage Perl écrit par Rasmus Lerdorf pour analyser les visites de son CV en ligne. Il réécrit l'application en langage C puis ouvre son programme à la communauté du Libre. Ainsi, il pourra désormais compter sur l'aide de nombreux développeurs. Son application PHP fusionne avec un moteur de traitement de formulaires, FI, ce qui donne la version 2. Avec la version 3, les fonctions deviennent nombreuses et la popularité grandit.

1.2 Evolution du langage

Le langage PHP a encore beaucoup évolué. Nous en sommes actuellement à PHP 6. Les usagers ne sont pas obligés de prendre le train à chaque avancée technique pour plusieurs raisons :

- De nombreux sites en production sont faits en PHP 3 ou PHP 4 et fonctionnent très bien ;

- Les hébergeurs de sites proposent souvent une configuration PHP 4 ;
- Le langage objet demande un raisonnement plus élaboré.

Plusieurs raisons peuvent pousser à changer de version :

- Une plateforme ou framework passe à cette version et nous intéresse ;
- Par défi et goût d'apprendre de nouvelles syntaxes ou cadres de travail .

Reportez-vous au chapitre intitulé *Les frameworks ou cadriciels PHP* pour obtenir plus de détails sur le sujet.

1.3 Intérêt de ce langage

PHP est libre, portable et facile à comprendre. C'est un langage pour tous. Ce n'est pas un langage ésotérique destiné uniquement aux professionnels, avec un jargon incompréhensible. Certains langages dits complexes sont à la mode car leur complexité permet à des entreprises de passer beaucoup de temps sur des applications cher payées. Le langage PHP est suffisamment souple, robuste et rapide pour intéresser les professionnels et sa richesse d'outils en fera, dans les années qui viennent, le langage privilégié de toutes les entreprises.

La liberté

PHP est gratuit. Ainsi, la plupart des hébergeurs gratuits le proposent avec la base de données *MySQL*, gratuite elle aussi. Libre signifie également que toute personne peut l'enrichir, à condition d'en faire profiter tout le monde gracieusement. À l'intérieur de la communauté du Libre, les licences déclinent des droits d'auteur distincts. En effet, Libre ne signifie pas que tout est permis ni que tout est gratuit.

Le langage PHP évolue constamment grâce à une communauté de 200 développeurs environ. La documentation est continuellement mise à jour et des milliers de sites à travers le monde s'y consacrent en publiant des articles, des forums, des astuces ou en proposant des applications gratuites. Le langage PHP est porté par une vague de popularité qui ne cesse de croître. Nous espérons à travers cet ouvrage participer modestement à son essor.

La communauté du logiciel libre (communauté du Libre)

En 1969, les chercheurs ressentent le besoin de partager leurs travaux. Ils créent donc un réseau de machines. Le système d'exploitation Unix est inventé la même année. Il définit l'ordinateur comme un nœud de communication. Ce système fonctionne sur tout type de machine. Les deux inventions constituent les fondations d'Internet.
La coopération engendrée par ce réseau de chercheurs accélère le rythme des découvertes et des inventions.
Vers la fin des années 70 se créent les grandes entreprises de logiciels. Le droit sur le logiciel apparaît, posant que le logiciel appartient à celui qui l'a créé. L'auteur (individu ou société) a le droit exclusif de le copier, de le distribuer et de vendre une licence à un utilisateur. La licence donne à l'utilisateur le droit d'installer le programme sur sa machine, de l'exécuter et d'en faire une copie de sauvegarde.
Les éditeurs de logiciels dissimulent le code source pour empêcher quiconque de se servir de leurs connaissances et d'en faire un produit concurrent. Ainsi, une guerre des standards se déclenche avec, à la clef, des profits extravagants. Dans cette course, seuls les plus gros survivent, ce qui nuit à la concurrence et rend le marché dépendant de certains produits.

Un chercheur qui partageait ses découvertes, Richard Stallman, s'aperçoit que ses connaissances sont utilisées par des entreprises qui vendent leurs produits sans qu'il en reçoive un quelconque retour. Il imagine alors la GPL (General Public License). Cette licence accorde à l'utilisateur le droit de diffusion, d'utilisation et de modification. Une clause supplémentaire donne l'obligation d'utiliser la même licence, que le logiciel soit modifié ou non.
En 1991, le Finlandais Linus Thorvalds invente Linux, un système d'exploitation libre et gratuit. Le combat actuel contre le brevet logiciel a lieu dans ce contexte car son application signifierait, à terme, la mort du Libre.

La portabilité

PHP peut être exécuté sur n'importe quelle machine. Les programmes que vous créerez dans ce livre sont exécutables sur Unix, Linux, Windows ou Macintosh. Cette caractéristique vous affranchit des décisions prises par Microsoft ou Apple et sécurise l'avenir de votre application.

La facilité

PHP peut être abordé par tous. D'une part, il est gratuit et enrichi continuellement par de nombreux développeurs (200 environ). D'autre part, il est simple de commencer à programmer en langage PHP. Tout cela joue un grand rôle dans sa popularité. Le fait qu'il soit soutenu par un grand nombre de développeurs assure une évolution continuelle et favorise l'intégration des dernières évolutions en matière de langage informatique.

Ce langage est simple. Vous commencez et obtenez immédiatement un résultat. Cela permet à chacun d'apprendre à son rythme (comme vous le verrez à travers ce livre) et d'intégrer la connaissance des fonctions et extensions au gré de ses envies.

Un peu d'histoire

Le langage PHP est issu principalement du langage Perl, il s'appuie donc sur le type `string`, ou chaîne de caractères. Le string en anglais est le fil d'un collier de perles. Dans le langage C qui est l'ancêtre de la plupart des langages utilisés actuellement, il n'existe pas de type `string`. Seuls existent les caractères comme `A` ou `b`… Une phrase est alors un tableau de caractères avec le premier caractère dans la première case et le chiffre `0` dans la dernière case, pour terminer la chaîne. Une `string` est un collier de caractères, ce qui est assez éloigné de la manière dont nous voyons les mots qui sont de vraies unités du langage. Le langage Perl a été conçu pour simplifier les règles qui présidaient à l'écriture de langages de programmation. Le principe était de rapprocher l'écriture du code du langage naturel, c'est-à-dire de la manière dont l'humain s'exprime. Le langage Perl a donné naissance au langage PHP tandis que d'autres langages ont suivi d'autres routes.

Le langage PHP, à partir de sa base Perl et C, a emprunté la conception orientée objet au langage Java (dans la version 3). Le principal objectif d'un langage objet est la réutilisation de briques de programmation autonomes et portables dans plusieurs programmes. Ce langage est dérivé du code C++, donc du langage C. La démarche des concepteurs de Java procède de la même constatation : le langage C++ est lourd à manipuler, il fallait donc le simplifier et en sécuriser l'utilisation. Les concepteurs créent le langage Java sous l'égide de Sun Microsystems. Le

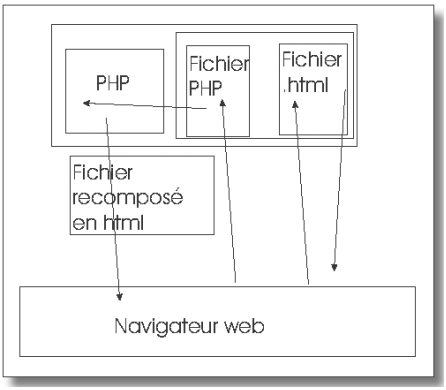
code Java n'a pas la souplesse du langage PHP, c'est pourquoi un morceau de programme en langage PHP qui prend quinze minutes à concevoir et à écrire demandera plus d'une heure en langage Java. Les programmeurs Java disent du langage PHP que c'est un langage de "bidouille". On peut effectivement faire tout et n'importe quoi. Ce que les programmeurs en langage PHP prennent comme un compliment. Les deux langages tendent à se rapprocher, du moins, le langage PHP peut maintenant, grâce à des bibliothèques, utiliser des classes Java. Le langage PHP est métissé de toutes ces cultures.

Dans ce livre, nous n'aurons pas la place d'aborder l'aspect programmation objet du langage PHP, le but de cet ouvrage étant de vous faire entrer directement dans le langage PHP pour résoudre n'importe quel problème posé à un programmeur. Par contre, dans l'articulation des fichiers PHP (l'architecture de l'application), nous nous inspirerons de la programmation objet. Il vous suffira ensuite d'apprendre la syntaxe et de pratiquer PHP pour maîtriser cet aspect du langage. La version 5 du langage PHP développera cet aspect. La caractéristique du langage PHP est de garder une compatibilité ascendante. Ainsi, certaines fonctions ont changé de nom mais l'ancien nom est toujours opérationnel. Tout cela fait du langage PHP un langage polymorphe. Nous le comparerons à une encyclopédie. Pour chaque problème posé, il existe plusieurs approches. Vous pouvez utiliser tout ce qui fonctionne à partir du moment où vous gardez une cohérence et une rigueur. N'oubliez pas en effet d'être rigoureux car le langage PHP ne vous obligera pas à suivre des chemins balisés.

Reportez-vous au chapitre **Mettre de l'ordre** pour plus de détails sur l'architecture, la rigueur et la clarté de votre programme.

1.4 Fonctionnement

Le noyau PHP est un moteur en langage C localisé sur le serveur. Quand un fichier est appelé par un navigateur web, selon son extension (*.htm*, *.php*, *.asp*…), le serveur l'envoie directement au navigateur pour qu'il affiche la page ou le traite (dans le cas d'un fichier *.php*) en exécutant les commandes du langage PHP dont le résultat est renvoyé sous forme de code HTML.



▲ Figure 1.1 : *Traitement d'un fichier PHP*

Le langage PHP est interprété dans sa version 3 et compilé à la volée dans la version 4. Le langage compilé est un peu plus rapide pour la machine.

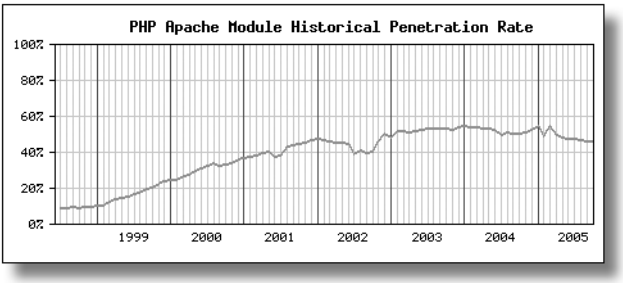
Langage C

C est un langage de haut niveau, compilé pour devenir exécutable sur une machine. Le langage naturel est celui que nous employons dans ce livre — du moins nous l'espérons. Le langage de haut niveau est un langage de programmation qui utilise des mots, le langage de bas niveau est quant à lui composé de 0 et de 1, seul langage compréhensible par une machine. C emploie une syntaxe et des mots qui sont ensuite transformés par un compilateur en une suite de 0 et de 1, selon le code de la machine et son système d'exploitation. Il existe des compilateurs C pour Linux ou Windows. Ainsi, un programme compilé sous Windows ne peut être utilisé sous Linux, et vice versa.

1.5 Des chiffres.

En février 2002, le serveur le plus répandu était Apache, avec 65 % des serveurs, suivi par Microsoft à 25 %. Selon l'enquête d'E-Soft (`http://www.securityspace.com`), dans les serveurs Apache, le module le plus répandu était le langage PHP avec 1 646 792 modules, c'est-à-dire qu'il

était installé sur environ 45 % des serveurs Apache. Ces chiffres ne tiennent pas compte du cas où le moteur PHP est installé sur un serveur Microsoft.



▲ Figure 1.2 : *Taux de pénétration du module PHP sur Apache*

En général, PHP est un langage de script, c'est-à-dire qu'il est inclus dans la page HTML. Il peut aussi être utilisé pour envoyer des commandes système à la machine ou pour créer des logiciels comme ceux que vous avez sur votre PC, grâce à l'outil PHP-GTK dont la version 0.5.0 vient de sortir. Elle était très attendue car c'est la première version stable. Elle ouvre de grandes perspectives pour les développeurs. Les impatients l'ont appelée "Lundi commence samedi".

2

Ma première page PHP

Nous commencerons par une revue sur le langage de description de page : HTML. Nous en évoquerons les principales balises. En fin d'ouvrage, vous trouverez toutes les références pour approfondir ce langage. Pourquoi parler de HTML dans un chapitre intitulé *Ma première page PHP* ? HTML est indispensable dans une page destinée à Internet, que l'extension de son nom de fichier soit *.asp*, *.php* ou *.jsp*… Une fois les balises HTML connues, nous aborderons le langage PHP.

2.1 HTML

L'ancêtre de HTML est SGML (Standard Generalized Markup Language). De ce premier langage très complexe sont issus HTML et XML. Ce métalangage a introduit pour la première fois les marqueurs ou balises. Le principe des balises est un concept intéressant car tout document peut être créé sur le plus simple des traitements de texte (Notepad sur Windows ou SimpleText sur Macintosh) et toutes les indications de mise en forme sont lisibles en clair dans un traitement de texte, tandis qu'elles sont interprétées par le navigateur internet sous forme de présentation multimédia.

Métalangage

Méta, en grec ancien, signifie "ce qui dépasse ou englobe". Un métalangage est un langage autour du langage. HTML décrit le langage avec des balises ou marqueurs qui n'apparaissent pas dans l'affichage du texte mais agissent sur la présentation de ce dernier.

Le langage HTML (Hypertext Markup Language) est accessible à tous. Il est composé de balises c'est-à-dire de codes encadrés de chevrons. Voyez la balise comme un ordre ponctuel envoyé à l'exécutant, le navigateur, qui concerne la présentation et la structure du document. Ces balises peuvent être solitaires ou accompagnées, c'est-à-dire composées d'une balise ouvrante et d'une balise fermante. Les règles de ce langage sont assez souples et les navigateurs internet programmés pour tolérer un certain nombre d'inexactitudes dans le code. Une fois que vous aurez appris les balises essentielles comme nous le ferons dans ce livre, vous pourrez approfondir ce langage qui, avec la version HTML 4, comprend une centaine de balises.

 est une balise solitaire alors que <title>texte ici</title> est composé d'une balise ouvrante et d'une balise fermante, encadrant un texte qui sert de titre au document.

HTML est donc un langage de description de page matérialisé par des balises. Il est interprété directement par le navigateur mais n'est en aucun cas un langage de programmation car il ne comporte pas de fonctions.

Voici un exemple de page simplissime :

Listing 2.1 : *Page simplissime : simplissime.html*

```
<HTML><HEAD><TITLE>Ma Page</TITLE></HEAD>
<BODY>Ma page très perso</BODY></HTML>
```

Pour que l'accent dans le mot `très` soit reconnu, il est nécessaire d'ajouter au début de la page, dans l'en-tête, `<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`, pour expliquer au navigateur qui ne serait pas configuré pour le français, que ce sont des caractères européens de type iso-8859-1 dont le français fait partie. Autrement, il suffira d'écrire cette lettre accentuée sous la forme `è`.

2.2 Le cheminement d'un apprenti webmestre

Prenons une page parmi les millions de pages sur Internet, une page HTML sur un site quelconque. La page est statique, elle restera identique tant que personne n'y touchera. Évidemment, elle est fréquentée par des visiteurs de passage. Elle sera enregistrée dans les favoris (ou signets) du navigateur de son auteur et de quelques visiteurs.

Un jour, l'auteur souhaite étoffer sa page puis ajouter d'autres pages avec des liens. Mais Internet est un flux d'informations constamment mises à jour, sa page restera momifiée et le temps rendra ses informations obsolètes.

L'auteur essaie de trouver des informations intéressantes à mettre à jour pour que sa page vive. Il modifie sa page pour en changer l'apparence, ce qui se révèle fastidieux, d'autant qu'il fait parfois de fausses manœuvres et éborgne une balise ou deux, entraînant des problèmes de présentation.

À son crédit, nous pouvons avouer qu'il n'est pas facile de mettre à jour une page dans un traitement de texte, tant le contenu se trouve constellé de balises. Il faut commencer par afficher le code source de votre page favorite. Pour cela, il suffit de choisir *affichage/source* dans le navigateur.

L'auteur ressent bientôt l'envie de créer un club. Ainsi, les membres pourront participer au contenu de son site. Mais il ne peut leur confier ses pages HTML pour qu'ils y ajoutent directement les informations. C'est là qu'il découvre dans la FAQ de son hébergeur qu'il dispose d'un moteur PHP et d'une base de données *MySQL*. Il lit qu'ainsi, les visiteurs peuvent participer à la vie du site en donnant des informations. Ces informations sont classées et stockées dans la base de données puis affichées grâce au langage PHP.

Il pourrait créer le club des Bons Vivants, parce qu'il aime manger, boire et faire la fête. Il pourrait ainsi organiser des repas, se faire de nouveaux amis et échanger recettes et adresses de bonnes caves dont les membres du club profiteraient. Mais si trop de monde connaît la bonne adresse, les prix risquent de monter et le produit de se raréfier. Il s'apprête donc à créer une sorte de club Privilège.

Il regarde de plus près le langage PHP, trouve un manuel, consulte des milliers de pages de documentation en ligne et commence à se décourager. Dans la FAQ, l'hébergeur lui dit pourtant que c'est simple. Mais par où commencer ? Il arrive finalement en ville et trouve le livre qui va lui ouvrir la porte de cet univers — nous espérons que pour vous ce sera celui-ci. Toute proportion gardée, c'est ainsi que l'aventure a débuté pour nous.

Vous avez maintenant compris l'intérêt du langage PHP et les raisons de sa popularité.

2.3 Quelques notions de HTML

Commencez par créer un fichier *index.html*. Toute page HTML est contenue dans les balises `<HTML></HTML>`. Les codes peuvent apparaître indifféremment en capitales ou en bas de casse `<html></html>`. Ensuite, la page est divisée en deux parties : l'en-tête et le corps.

L'en-tête contient des éléments invisibles, des méta-informations sur la page : son titre et le type de caractères qu'elle emploie.

```
<HEAD><TITLE>Mon titre</TITLE><meta http-equiv="Content-Type"
 content="text/html; charset=iso-8859-1"></HEAD>
```

Ces éléments sont obligatoires ou conseillés. D'autre éléments, optionnels, peuvent apparaître dans l'en-tête.

En dessous, la partie affichée dans la fenêtre du navigateur, le corps, est délimitée par les balises <BODY></BODY>.

La page index ou default

La page index est la page appelée automatiquement dans un répertoire si aucun fichier n'est spécifié. Ainsi, l'adresse *http://www.pourpre.com* est en réalité l'abréviation de *http://www.pourpre-.com/index.htm*, *index.html*, *default.htm*, *default.html* ou *index.php*…, c'est-à-dire que l'URL appelle le fichier *index.htm* par exemple, dans le répertoire *htdocs* ou *www* du serveur *pourpre.com*. Ces deux noms de répertoire peuvent qualifier le répertoire public internet où toute requête extérieure vers le serveur sera dirigée. Il est conseillé de mettre ce fichier dans tout répertoire.

Les éléments qui structurent un document sont les paragraphes, avec les balises <P> et </P>. Pour une raison de compatibilité descendante, la balise <P> à la fin ou au début d'un paragraphe suffit à créer un nouveau paragraphe, sans la balise de fin. La balise <P> peut comporter les attributs du paragraphe. Tout le texte inséré entre les balises et sera aligné à gauche. Notre but n'est pas de présenter un panorama des balises HTML mais de vous fournir les balises essentielles pour composer une belle page. Ainsi, vous rencontrez aussi la balise <P> </P> ou . Les alignements de paragraphes sont caractérisés par les attributs "left" pour gauche, "right" pour droite et "center" pour centre. HTML est polymorphe. À chacun de choisir ensuite la syntaxe qui convient le mieux ou celle que son logiciel de création de page lui propose.

les espaces HTML

Les espaces ne sont pas significatifs. Ainsi, un double espace n'affichera en réalité qu'un espace. Vous pouvez inclure des espaces insécables avant un point d'interrogation par exemple, pour qu'il ne soit pas isolé sur une autre ligne. Ce code est ` `. N'en abusez pas, certains logiciels en raffolent.

Quant aux polices de caractères, vous pouvez les caractériser avec la balise `<font></font>`. Les attributs sont `color`, `face`, et `size`. Comme dans l'exemple ci-dessous :

```
<font size="3" face="Arial, Helvetica, sans-serif" color="#FF6633">
```

`size` peut être une valeur absolue comme 1, 2, 3… ou une valeur relative +1, +2 ou −1,−2. La valeur relative attribuera une valeur au-dessus ou en dessous de la taille de la police du paragraphe.

`face` proposera des noms de police. Si cet attribut comprend plusieurs polices, le navigateur cherchera la première. Si elle est disponible dans son système, il l'utilisera, sinon il cherchera la seconde. Dans le cas où aucune police proposée n'est disponible, il utilise une police par défaut.

Les polices propres à Macintosh

Dans la liste des polices affichées par l'attribut `face`, indiquez au moins une police de caractère spécifique au Macintosh, `geneva` par exemple.

`color` peut recevoir un attribut de couleur par nom ou valeur, par exemple `blue` ou `#0000FF`. Tous deux sont synonymes ; la première valeur est un nom, la seconde est le code hexadécimal qui lui correspond dans la gamme. Cette valeur est composée des codes RVB (ou RGB en anglais, c'est-à-dire Rouge-Vert-Bleu) de la couleur choisie.

Les valeurs hexadécimales

Nous avons l'habitude de compter par dix. Nous comptons de 0 à 9 puis à 10, revenons à 0 en ajoutant une unité de dizaine. Il s'agit d'une base 10. Les Celtes comptaient par vingt, c'est pourquoi les nombres quatre-vingt ou soixante-dix ont subsisté en français, excepté en Suisse et en Belgique. Le compte hexadécimal utilise une unité de 16. Comme nous ne possédons pas de chiffres au-delà du 9, il utilise des lettres. (0..9 puis A, B, C, D, E, F pour les valeurs 10, 11, 12, 13, 14, 15. Ainsi le nombre 9889598 devient 96E73E et permet de gagner une case mémoire. Ce système permet d'utiliser seize millions de couleurs (256^3) avec seulement six caractères.)

La couleur sert également à caractériser le fond de la page ou celui du tableau.

Ainsi, dans la balise <BODY BGCOLOR="#FFFFFF">, la couleur blanche est attribuée au fond de page. Elle est aussi la couleur par défaut. Si vous l'omettez, vous obtiendrez un fond blanc ou noir selon le navigateur, une police noire, des liens hypertextes en bleu…

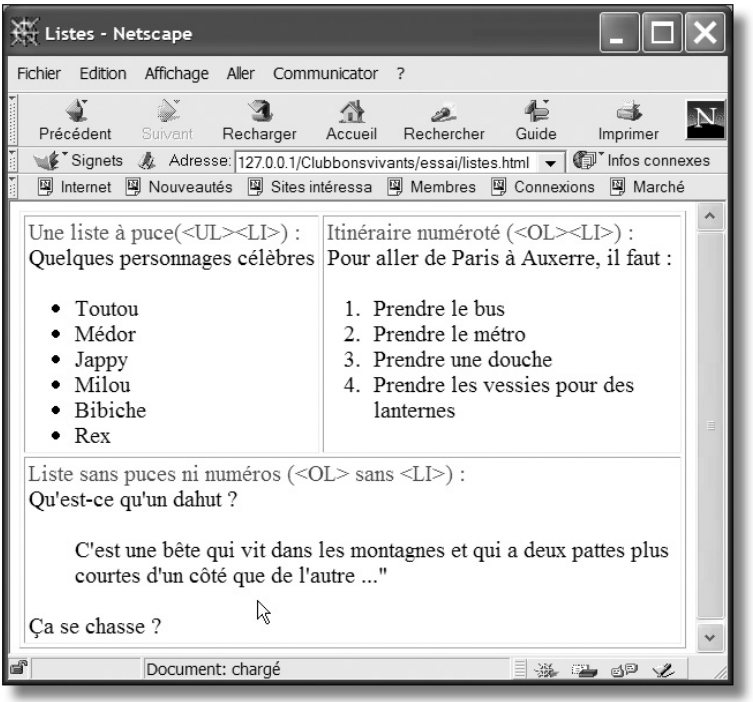
N'oubliez pas le signe dièse

Le signe dièse précède tout code de couleur, comme dans #FFFFFF.

Les paragraphes comportent parfois des titres. Du plus grand au plus petit, vous avez les balises <H1>, <H4> avec leur balise de fermeture.

Le paragraphe peut inclure des listes numérotées. La balise qui caractérise une liste numérotée est <OL></OL>. Chaque ligne, à l'intérieur, est insérée dans un <LI></LI>. Ainsi, chaque ligne sera automatiquement numérotée.

<UL></UL> caractérise une liste à puces. Ces balises de listes permettent aussi d'ajouter un retrait à la ligne ou au paragraphe, en omettant les <LI></LI>.



▲ Figure 2.1 : *Liste à puces, liste numérotée, liste sans balise <LI>*

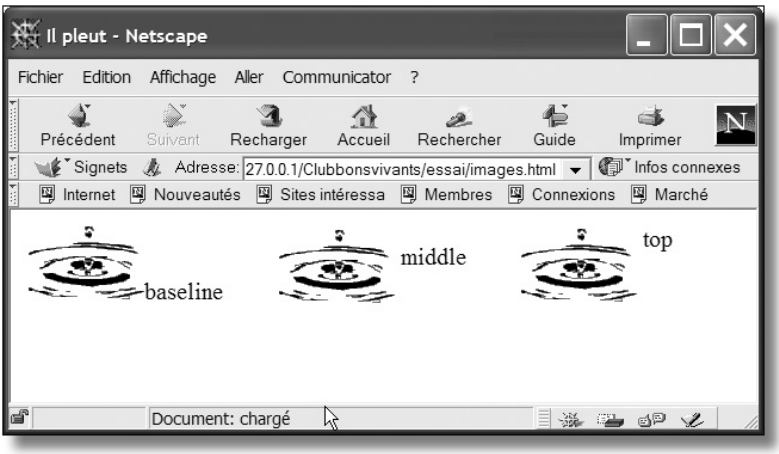
Les images

Les principaux formats du Web sont *GIF* et *JPEG*. La compression GIF est intéressante si votre image contient des aplats de couleur, un logo par exemple, la compression JPEG est plus performante sur une photo qui comprend de nombreuses nuances. La compression a pour effet d'alléger le poids d'une image en kilooctets, donc de l'afficher plus vite dans la fenêtre du navigateur.

Le format *GIF* permet également de sélectionner une couleur pour la rendre transparente, ce qui aura pour vertu de faire apparaître la couleur de fond de page en lieu et place de la couleur choisie. Si par exemple, vous avez un logo rond, il est dommage de voir ce logo apparaître dans un carré

blanc. En rendant son fond transparent, cette image apparaîtra ronde, posée sur le fond de page, quelle que soit sa couleur.

La balise pour intégrer une image dans une page est `<img src="images/image.gif" width="100" height="100" alt="liste des candidats" align="bottom">`. Pour que l'image s'affiche plus vite, il est préférable d'ajouter la largeur `width` et la hauteur `height`. L'image est souvent utilisée comme lien de navigation, il est donc conseillé d'y ajouter l'attribut `alt` qui affichera un texte dès que le curseur de la souris la survolera. Ce texte explique la destination du lien, ou le cas échéant, attribue une légende à l'image. L'attribut `align` place le texte qui l'accompagne. Dans le cas de `bottom`, le texte se place en bas de l'image, `middle` place le texte au milieu et `top` en haut de l'image.



▲ Figure 2.2 : *Bottom, middle, top*

Les liens

Vous avez une page, il faut maintenant la relier à d'autres pages. Pour cela, vous utilisez les liens hypertextes.

Le lien se fait par l'intermédiaire d'un mot ou d'une image, vers une autre page ou un autre paragraphe de la même page :

```
<A HREF="autrepage.htm"> mon autre page</A>
<A HREF="autrepage.htm">
<img src="images/image.gif"  width="100" height="100"
 alt="liste des candidats" align="bottom"></A>
```

L'attribut border dans l'image

Il est préférable de toujours ajouter l'attribut `border="0"` dans la balise d'insertion d'image pour que l'image ne soit pas encadrée par un liseré parasite de la couleur du lien.

Le lien vers un paragraphe à l'intérieur d'une page utilise la même syntaxe. Ce type de lien est pratique quand une page est longue. Une série de liens, en haut de la page vers les différentes parties de la page, facilite la navigation :

```
<A HREF="#conclusion">conclusion </A>
```

Dans cette partie de la page, vous inscrirez l'ancrage `<A NAME="#conclusion">quelques mots</A>`. L'URL complète, pour parvenir directement à ce paragraphe à partir d'une autre page sera :

```
<A HREF="autrepage.htm#conclusion">Conclusion du colloque</A>
```

Un autre paramètre de lien, `target`, affiche le résultat du lien dans une fenêtre choisie. Le paramètre le plus couramment utilisé est `_blank` qui ouvre une nouvelle fenêtre dans le navigateur pour afficher la page cible du lien. Cette fenêtre permet de garder présente la page appelante :

```
<A HREF="parcours.htm" target="_blank">Comment s'y rendre</A>
```

Le dernier type de lien est celui qui englobe une adresse e-mail :

```
<A HREF="mailto:moi@moi.com">moi@moi.com</A>
```

Ce lien cliqué ouvrira le logiciel de messagerie par défaut avec, comme destinataire, l'adresse e-mail en question.

Formater un courriel dans le lien de l'adresse e-mail

Vous pouvez ajouter divers éléments dans le message en incluant un point d'interrogation à la fin de l'adresse e-mail et une esperluette pour séparer les champs.

```
<A HREF="mailto:moi@moi.com?subject=test&Cc=toi@moi
.com&Body=Salut">
écrire</A>
```

La taille des données y est limitée.

La couleur des liens sera déterminée dans la balise <BODY> par les paramètres `link`, `vlink`, `alink` suivi de la couleur. `link` donne la couleur du lien, `vlink` la couleur du lien pour un site déjà visité et `alink` (`active link`) la couleur lorsque vous cliquez dessus.

```
<body bgcolor="#FFFFFF" text="#000000" link="#0000FF"
 vlink="#993399" alink="#FF0000">
```

Les tableaux

Les tableaux sont indispensables pour structurer une page. La plupart du temps, ils sont invisibles mais permettent de placer les images et le texte avec plus de précision dans la surface de la page.

Le tableau est contenu dans les balises <TABLE></TABLE>. Il est formé de rangs et de colonnes.

Les rangs sont créés à l'aide des balises <TR> et les colonnes, avec des balises <TD> :

```
<TABLE><TR><TD></TD><TD></TD></TR> </TABLE>
```

Ici, vous avez une table d'un rang et de deux colonnes. Sur un des rangs, vous pouvez fusionner les deux colonnes. Pour cela, utilisez un attribut <TD COLSPAN="2"> :

```
<TABLE><TR><TD></TD><TD></TD></TR>
<TR><TD COLSPAN="2"></TD</TR>
</TABLE>.
```

Vous pouvez aussi faire la transformation sur deux cellules dans une même colonne, sur deux lignes adjacentes :

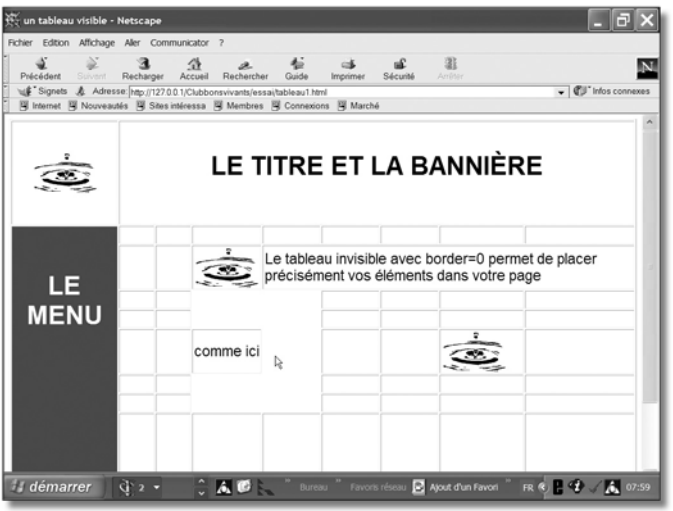
```
<TABLE><TR><TD ROWSPAN="2"></TD><TD></TD></TR>
<TR><TD></TD></TR>
</TABLE>
```

Pour voir l'effet de votre code dans la fenêtre du navigateur, nous vous conseillons d'ajouter `border="1"` dans la balise table `<TABLE border="1">`, le temps de faire le test. Ainsi, vous marquerez les limites de vos cellules de tableau avec une bordure. Exercez-vous. Tapez du code dans un traitement de texte comme Notepad (ou Bloc-notes) dans les accessoires de Windows ou Simpletext, dans le menu **Pomme** du Macintosh. Ensuite, nommez-le *index.htm* ou *index.html* et ouvrez-le dans votre navigateur. Vous touchez à la dimension magique de simples mots qui créent instantanément un univers visuel à l'écran.

Pour placer des éléments dans votre page au moyen d'un tableau invisible, il faut en calibrer la largeur. La taille de chaque colonne devra être précisée car autrement, le tableau adaptera chaque colonne au contenu de ses cellules, réduisant à rien les colonnes vides. Une colonne vide devra contenir un espace insécable ` ` pour que la taille puisse être prise en compte. La taille est déclarée soit par un nombre de pixels, soit par un pourcentage. L'intérêt d'utiliser le pourcentage, au moins pour le tableau complet, est que l'écran de vos visiteurs peut afficher des définitions différentes (640 pixels par 480, 1024 x 768 ou 1280 x 1024) et qu'une page structurée par un tableau déterminé en pourcentage s'adaptera à chaque configuration. Le pourcentage est une valeur relative à la taille tandis que le pixel est une valeur absolue.

`<TABLE width="95%">`. Ici, le tableau occupera tout l'écran ou presque, quelle que soit la taille de l'écran. Vous pouvez aussi colorer le fond de ce tableau avec `bgcolor="#FF00CC"` en le positionnant dans la balise `table`, `tr` ou `td`, selon la surface que vous souhaitez colorer.

▲ Figure 2.3 : *Tableau invisible avec fond*



▲ Figure 2.4 : *Tableau visible avec bordures*

Les formulaires

Pour que les visiteurs participent à votre site, il faut mettre à leur disposition un formulaire qui vous permette de recevoir une information structurée de leur part. Les visiteurs rempliront alors des champs déjà calibrés.

Les champs du formulaire seront contenus dans les balises <FORM> et </FORM>. La balise <FORM> ouvre le formulaire et contient obligatoirement les attributs `action` et `method`.

L'attribut `action` donne l'adresse internet (URL) du script qui traitera les données. L'attribut `method` indique la manière dont les données seront envoyées au programme. Avec la méthode `GET`, les paramètres renseignés dans chaque champ du formulaire sont envoyés dans l'URL :

```
http://www.monsite.com/traitement_formulaire.php?titre=Le
+Lait%2C+une+sacr%E9e+vacherie+%3F&auteur=Dr+Nicolas+Le+Berre
```

Une variable se crée automatiquement pour chaque champ délimité par une esperluette, cette URL équivaut à la création de variables avec des valeurs en haut du fichier appelé :

```
$titre='le Lait, une sacrée vacherie' ;
$auteur='Dr Nicolas Le Berre';
```

Les variables seront abordées dans le chapitre 3.

L'URL sera codée, ainsi le caractère littéral `?` est remplacé par `3F` pour éviter d'être confondu avec l'opérateur de transport des données `?`. De même, `é` est remplacé par son code hexadécimal `E9`. Ils sont précédés d'un signe d'échappement `%` qui indique au moteur PHP qu'il s'agit de code. Le signe `+` marque les espaces. Pour coder ce type de chaîne dans un lien et non plus dans un formulaire, il vous suffit d'utiliser la fonction `urlencode()`. Ensuite, les données n'ont pas besoin de fonction car le moteur PHP les décode automatiquement.

L'autre méthode est POST qui est plus propre et peut envoyer des données dont la taille va jusqu'à 64 Ko. Ces données ne sont pas visibles.

L'attribut name de chaque champ

Le nom de chaque champ baptise la variable PHP et sa valeur est attribuée à la variable ainsi créée. Le nom de chaque champ doit donc être cohérent avec vos propres conventions de nommage des variables.

Tab. 2.1 : Les champs INPUT

Champ	Caractéristique	Attributs
text	Pour entrer une information de type texte	`size` : longueur du champ affiché `maxlength` : nombre maximal de caractères entrés `value` : valeur par défaut
radio	Pour sélectionner une option, à l'exclusion des autres qui portent le même nom	Les différentes options ont le même nom (`name`) mais pas la même valeur (`value`). `checked` indique la case cochée par défaut.
checkbox	Pour sélectionner une ou plusieurs options	L'attribut `name` des cases est différent pour chacune. Dans l'hypothèse de cases multiples, les cases portent le même nom. `checked` indique la case cochée par défaut.
hidden	Pour transmettre une information qui ne sera pas modifiée par l'internaute, ni affichée	Comme pour un champ de texte, il comporte les attributs `name` et `value`.
submit	Pour transmettre les données d'un formulaire à un programme	Affiche un bouton d'envoi de formulaire.
reset	Pour vider les valeurs inscrites dans le formulaire	Affiche un bouton pour effacer le contenu du formulaire.
password	Pour entrer un mot de passe	Affiche le mot de passe sous forme d'étoiles.

Tab. 2.1 : Les champs INPUT

Champ	Caractéristique	Attributs
file	Pour envoyer un fichier dans le formulaire ou déposer un fichier dans le répertoire d'un serveur	Le bouton `parcourir` lui sera adjoint par le navigateur. Ainsi l'internaute ira chercher le fichier sur son disque dur.
image	Pour soumettre un formulaire avec une image au lieu d'un bouton	Remplace avantageusement le bouton `submit` par une image.

Tab. 2.2 : Les autres champs

Champ	Caractéristique	Attributs	
textarea	Pour entrer un texte sur plusieurs lignes Le texte inscrit entre les balises `<textarea>` et `</textarea>` est le texte affiché par défaut.	`rows` : hauteur de la fenêtre `cols` : largeur de la colonne `wrap` : découpage des lignes	
		`virtual` (conseillé)	Le texte est affiché sous forme de lignes mais envoyé en une seule ligne.
select	Pour sélectionner une ou plusieurs options dans une liste déroulante	`multiple` permet de sélectionner plusieurs options. `size` permet d'afficher une à plusieurs options dans la fenêtre.	
option (à l'intérieur des balises `select`)		`selected` indique l'option sélectionnée par défaut.	

Les cadres ou frames

Les cadres sont des fenêtres incorporées dans la fenêtre principale. Vous créez d'abord votre page d'accueil contenant des cadres qui appellent d'autres pages. Ainsi, vous pouvez afficher un haut de page avec une bannière, un logo ou un titre puis un menu de liens à gauche pour naviguer sur le site et enfin une fenêtre qui affiche le contenu des pages appelées par un clic.

L'intérêt de ce procédé est d'avoir un seul fichier d'en-tête et un seul fichier de menu, ce qui simplifie la mise à jour puisque toutes les pages appellent le même en-tête et le même menu. L'inconvénient, c'est que tous les navigateurs web n'affichent pas la page de la même façon, ainsi certaines pages présentent des bords rognés par le cadre. D'autre part, les moteurs de recherche trouveront une première page vide qui sert uniquement de structure pour afficher d'autres pages…

Nous préférons utiliser des tableaux. Il suffit, pour cela, de concevoir un gabarit de page pour la création des pages du site. Vous enregistrez cette page type sous un autre nom (*Fichier/Enregistrer sous*) pour créer un nouveau fichier et en modifiez le contenu pour produire une nouvelle page. Le menu peut être élaboré à partir d'une table de la base de données. La structure de la page, elle, est calquée sur un tableau invisible qui aide à disposer les éléments de la page.

Procédure à suivre

Créez votre page d'accueil que vous appellerez *index.htm* ou *index.html*. Contentez-vous de construire vos cadres et d'appeler les vrais pages. Procédez ainsi :

Listing 2.2 : *Des cadres : cadres.html avec répertoire cadres*

```
<html>
<head>
<title>Des cadres</title>
<meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1">
</head>
<frameset rows="80,*" cols="136,*" frameborder="NO"
 border="0" framespacing="0">
 <frame name="cornerFrame" scrolling="NO" noresize src="logo.html" >
 <frame name="topFrame" scrolling="NO" noresize src="titre.html" >
 <frame name="leftFrame" scrolling="NO" noresize src="menu.html">
 <frame name="mainFrame" src="page.html">
</frameset>
<noframes><body bgcolor="#FFFFFF" text="#000000">
votre navigateur n'accepte pas les cadres.
</body></noframes>
</html>
```

Vous disposez vos cadres dans la page en les dimensionnant. Vous avez deux unités de mesure : le pixel ou le pourcentage. Le pourcentage est préférable car vous ne connaissez pas les paramètres d'affichage des

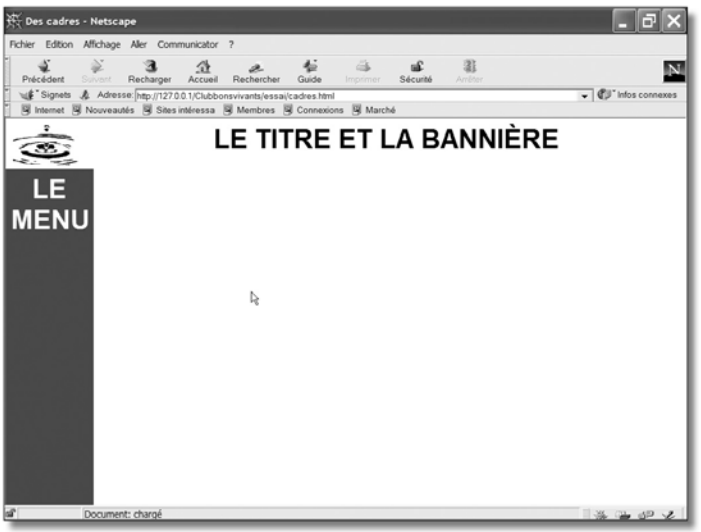
machines des internautes. L'étoile permet aussi d'indiquer que la taille dépendra du contenu.

Dans l'exemple, deux cadres sont créés à l'horizontale. Le premier possède une hauteur de 80 pixels et le second s'adaptera à son contenu (étoile). Dans le premier, nous avons placé deux cadres : un pour afficher le logo du site et un pour afficher une bannière. À l'intérieur du deuxième cadre horizontal, deux nouveaux cadres sont dessinés à la verticale. Le premier, *menu*, est large de 136 pixels, le second, *page*, s'adapte à la dimension restante.

L'attribut `NORESIZE` empêche le navigateur de changer la dimension du cadre et le paramètre `SCROLLING` caractérise la glissière, en bas ou à droite, sur laquelle une jauge coulisse pour révéler les parties cachées du cadre.

Chaque cadre a pour paramètres un nom donné par l'attribut `NAME` et l'URL de la page appelée par l'attribut `SRC` (abrégé de `source`).

Les balises `<noframes></noframes>` renferment un code qui s'affichera si le navigateur ne reconnaît pas les cadres, ce qui est de plus en plus rare.



▲ Figure 2.5 : *Résultat du code ci-dessus*

Les pages appelées sont créées, comme n'importe quelle page HTML :

```
<html>
<head>
<title>des cadres</title>
<meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1">
</head>
<body bgcolor="#663366" text="#000000">
<h1 align="center"><font face="Arial, Helvetica, sans-serif"
color="#FFFFFF">LE
  MENU</font></h1>
</body>
</html>
```

Remarquez le lien qui contient l'attribut `TARGET` (pour cible), avec le nom du cadre qui affiche le contenu, en l'occurrence le cadre *page*. Cette dernière caractéristique affichera la page demandée par le visiteur dans le cadre *page*, c'est-à-dire le cadre de droite.

2.4 Les outils

Bien qu'il soit possible, comme aux temps héroïques des débuts d'Internet, de créer vos applications avec un petit traitement de texte réduit à ses plus simples fonctions comme Notepad, vous disposez maintenant d'outils confortables dont certains sont gratuits. Ne vous en privez pas.

Dans cette partie, nous répertorions ce dont vous avez besoin pour l'écriture de vos applications, leurs tests et leur publication sur le Web.

Les logiciels d'édition de code PHP

Pour écrire vos pages HTML, vous pouvez employer un traitement de texte — le Web utilise surtout des fichiers texte. En langage PHP, si vos fichiers peuvent être simples au départ, vous vous apercevrez vite qu'ils deviendront illisibles en se complexifiant. Or les logiciels d'édition vont colorer votre syntaxe et insérer des retraits de ligne fort utiles pour vous repérer dans le code. Nous vous conseillons d'en acquérir un.

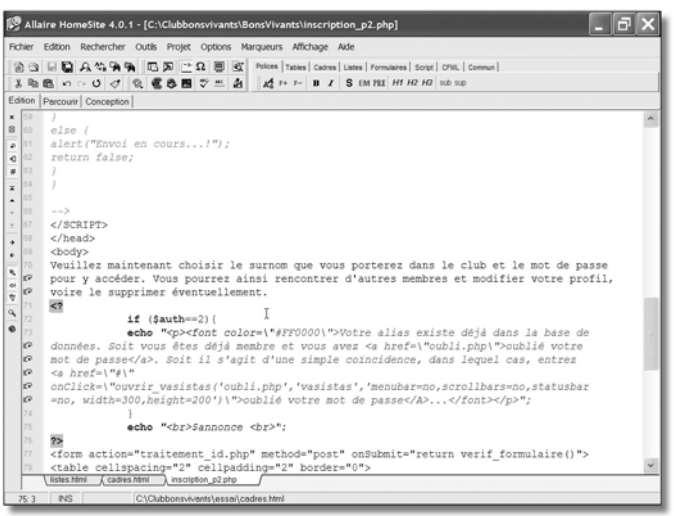
PHP Coder est un excellent logiciel gratuit. Il est en anglais et possède toutes les fonctions pour un bon confort de codage. Il colore la syntaxe,

comprend un interpréteur pour tester votre programme et permet d'annuler vos frappes sur de nombreux niveaux.

Cette dernière caractéristique nous a souvent permis de sauver du code. Ainsi, lorsque vous avez tapé des lignes que vous avez ensuite modifiées plusieurs fois. À une étape de votre application, vous vous apercevez que vous êtes en train de vous égarer et qu'il est plus sain de revenir au début pour prendre un autre chemin. Ce logiciel vous permet de revenir dans le passé et de faire un couper/copier de cet ancien code pour l'insérer ensuite dans le code présent.

Si vous préférez un logiciel en français, l'excellent HomeSite possède des fonctions identiques et bien d'autres. Il est livré seul ou avec un autre logiciel qui simplifie votre vie de webmestre et de créateur de pages internet, Dreamweaver de Macromedia. Pour Macintosh ou Linux, d'autres logiciels sont à votre disposition.

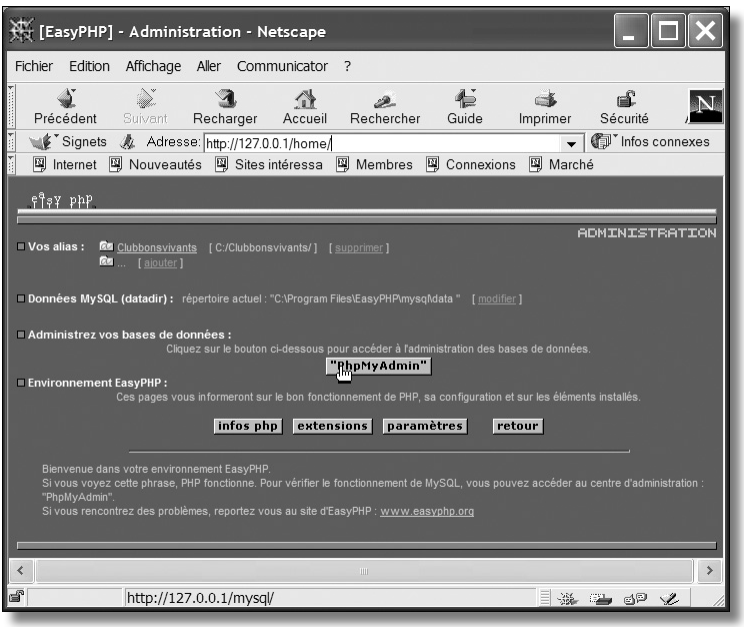
En annexe de cet ouvrage, vous trouverez une liste d'autres logiciels.



▲ Figure 2.6 : *HomeSite 4*

EasyPHP

Vous disposez de plusieurs installations gratuites, comme EasyPHP ou PHPTriad, qui incluent un serveur web Apache accompagné du module PHP et de la base de données *MySQL* pour votre machine personnelle sous Windows. L'intérêt de l'installer sur votre machine réside dans le coût et la facilité à tester vos programmes. D'autre part, vous prendrez le temps de concevoir l'architecture de votre programme avant de le mettre en ligne, prêt à l'emploi. À moins que vous n'ayez une connexion ADSL ou câblée, vous avez souscrit un forfait ou utilisé un fournisseur gratuit dont le temps de connexion vous est facturé. Vous avez donc intérêt à employer votre connexion à bon escient. Les tests et le débogage de vos programmes peuvent prendre beaucoup de temps, alors autant le faire sur votre machine, hors connexion. Pour mettre ensuite vos pages en ligne et les rendre visibles au public, vous aurez simplement à modifier quelques paramètres dans un fichier commun à toutes vos applications.

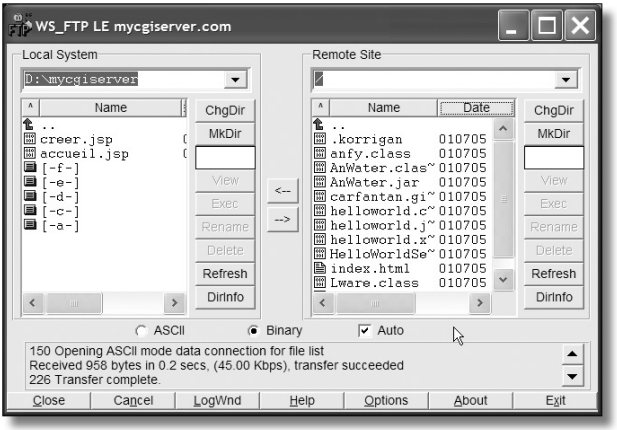


▲ Figure 2.7 : *EasyPHP*

Pour l'installation d'EasyPHP, la procédure est simple. Vous téléchargez le logiciel sur `www.easyphp.org` puis l'installez en un clic. Il installe le serveur Apache, le moteur PHP le plus récent et la base de données *MySQL* accompagnée de l'outil d'administration PHPmyAdmin. Sous Windows, vous avez déjà un serveur web installé, IIS. Si vous rencontrez un problème pour voir apparaître la page d'accueil d'EasyPHP, c'est souvent à cause d'un conflit avec le serveur de Windows, IIS. Il suffit alors de le désactiver. De même, des logiciels pare-feu comme ZoneAlarm, par ailleurs utiles, peuvent empêcher l'accès à la page du logiciel d'administration de la base *MySQL*, PHPmyAdmin. Dans la plupart des cas, EasyPHP ne pose aucun problème. D'autres installations comprenant PHP et *MySQL* existent, y compris pour Mac OS. Nous les citons en annexe de l'ouvrage.

Autres logiciels

Pour envoyer vos pages sur l'espace qui vous est alloué sur le serveur de l'hébergeur, utilisez un logiciel FTP dont vous trouverez la liste en annexe de cet ouvrage.



▲ Figure 2.8 : *WS_FTP*

La plupart de ces logiciels comportent une fenêtre à gauche pour votre machine et une à droite pour votre répertoire sur la machine distante.

Auparavant, vous aurez entré le nom d'usage (username ou login) et le mot de passe (password) octroyés par l'hébergeur. Dans le nom de domaine, vous entrez une URL ou un numéro IP. Entre les deux fenêtres, deux flèches montrent le sens de la transaction. Selon la provenance et la destination de vos fichiers, vous cliquerez sur l'une ou l'autre, après avoir sélectionné dans la fenêtre de provenance le fichier en question.

Pour Macintosh, le logiciel le plus courant est Fetch.

L'hébergement

Pour l'hébergement de vos pages, vous avez l'embarras du choix. Vous trouverez une liste en annexe du livre. Pour choisir, soyez attentif à plusieurs critères :

- La place de la publicité sur votre page ;
- La présence de la fonction `mail()` pour envoyer des e-mails à partir de votre programme PHP et d'un *htaccess* pour protéger vos répertoires ;
- La cohérence de votre adresse internet (URL), qui doit être courte pour être mémorisée facilement ;
- La plate-forme sur laquelle le site est développé, NT ou Linux/Unix.

PHP fonctionne bien sous Windows NT, mais vous ne disposez pas de toutes les fonctionnalités de Unix, en particulier pour les fonctions concernant le système de fichiers.

Reportez-vous au chapitre *Des algorithmes, des outils et des fonctions* pour plus de détails.

Le fichier *.htaccess* est un fichier sous Unix qui protège certains répertoires par une authentification.

Reportez-vous au chapitre *L'internaute est d'abord une personne* pour plus de détails.

Allez sur différentes pages de l'hébergeur sélectionné et interrogez d'autres webmestres pour savoir s'ils sont satisfaits des services proposés.

Un hébergeur francophone

Vous parlez mal l'anglais, préférez alors un hébergeur francophone. En effet, le jour où vous avez des problèmes et devez passer par le soutien technique, vous serez content de décrire votre souci en français.

Vous pouvez changer en cas de déception. Une fois l'URL de votre site diffusée, il est cependant fastidieux, voire parfois impossible, de contacter toutes les personnes pour les avertir du changement. Placez alors une page de redirection chez votre ex-hébergeur pour aiguiller les visiteurs vers votre nouvelle page.

La plupart des hébergements gratuits utilisent un serveur Apache installé sur des machines sous Linux. Ces offres gratuites sont rendues possibles par le monde du Libre.

Reportez-vous à l'annexe pour obtenir la liste des offres d'hébergement.

2.5 Ma page

Maintenant que vous avez installé les logiciels, vous allez pouvoir entreprendre de petites choses et entrer réellement dans le monde des programmeurs.

Ouvrez votre traitement de texte ou d'édition de code. Entrez les balises obligatoires `<HTML></HTML>`. À l'intérieur, ajoutez les balises qui composent les deux parties d'une page : l'en-tête `<HEAD></HEAD>` et, au-dessous, les balises `<BODY></BODY>`. Vous obtenez une page blanche. Maintenant, ajoutez un titre dans les balises `<HEAD>`qui apparaîtront comme une enseigne dans l'en-tête du navigateur. `<TITLE>Le club</TITLE>`.

La page est encore vide, alors vous allez vous présenter. Dans l'étau des balises `<BODY>`, écrivez `je m'appelle`.

Tapez maintenant votre premier code PHP.

Pour insérer du code PHP dans une page, vous devez l'entourer des balises <? ?> (ou parfois <?php ?> ou <% %>).

Avant la balise <HTML>, vous allez insérer votre première variable.

```
<?
$mon_nom="François";
?>
```

Vous remarquez le signe dollar, l'espace souligné (underscore), le signe égal, les guillemets et le point-virgule.

Le signe $ initiant un mot indique qu'il s'agit d'une variable. Une variable est un espace de mémoire vive alloué à une information. On préfère lui donner un nom compréhensible. `mon_nom` signifie davantage qu'une adresse mémoire avec du code informatique. Quand vous relirez votre programme, vous saurez qu'il s'agit du nom de quelqu'un, en l'occurrence de celui de François Rabelais car vous êtes un bon vivant et aimez la bonne chère. Ici, l'espace souligné remplace l'espace car on ne peut utiliser d'espace dans une variable.

Le signe égal crée une affectation, les guillemets indiquent au moteur PHP qu'il s'agit ici d'une chaîne de caractères et le point-virgule ferme la commande. La commande peut s'étaler sur plusieurs lignes mais le point-virgule indique au moteur que ce qui a précédé est une unité, un ordre d'exécution. L'affectation est l'action d'affecter une valeur à une variable. Ainsi, l'espace mémoire appelé `mon_nom` reçoit la valeur `François`.

Le signe égal

Le fait que les premiers concepteurs de programmes aient choisi le signe égal pour caractériser l'affectation d'une valeur à une variable entraîne souvent la confusion avec l'égalité qui, elle, est signalée par un signe double égal (==). Certains bogues sont dus à cette confusion, surtout dans les tests avec `if` que nous verrons dans le chapitre 3.

Enregistrez maintenant votre page dans le répertoire *club* à l'intérieur du répertoire *www* du dossier d'EasyPHP sous le nom *index.php*. Puis ouvrez votre navigateur web et faites un clic avec le bouton droit de la souris sur la petite icône dans la barre de tâches de Windows qui symbolise EasyPHP. Un menu apparaît. Sélectionnez **Web local** et cliquez sur le petit répertoire *club*. Rien n'apparaît, juste une page blanche. Vous avez créé une variable mais ne l'avez pas envoyée sur l'écran. Pour cela, inscrivez `<? echo $mon_nom ?>` en dessous de la balise `body`. Vous remarquez que nous avons omis le point-virgule. En fait, la balise `?>` inscrit la fin de la commande et le point-virgule est ici optionnel.

Maintenant, vous avez plusieurs méthodes pour ajouter le nom.

Tab. 2.3 : Tous les chemins mènent à Rome

Première ligne	Deuxième ligne
`$mon_nom="François Rabelais";`	
`$mon_nom="François"." "."Rabelais";`	
`$mon_nom = "François";`	`$mon_nom =$mon_nom." Rabelais";`
`$mon_nom = "François";`	`$mon_nom="$mon_nom Rabelais";`
`$mon_nom = "François";`	`$mon_nom.=" Rabelais";`

Ces cinq solutions donnent le même résultat, `François Rabelais`.

Passons maintenant aux explications.

La ligne 1 est une chaîne de caractères. Vous écrivez comme d'habitude, en langage naturel, avec un espace entre les mots.

La ligne 2 du tableau introduit une fonction exprimée par le point, une fonction de concaténation. Ainsi, cette fonction assemble les trois tronçons de phrase, espace inclus. Les guillemets doubles comme les guillemets simples indiquent que la valeur est une valeur de type `string`. Dans certains langages, vous devez toujours préciser le type d'une donnée et il est malaisé de faire passer une donnée d'un type à un autre, ce qui n'est pas le cas en langage PHP qui reconnaît de lui-même le type de la donnée. Nous verrons cela dans le chapitre 3.

Il existe des différences entre le guillemet double et le guillemet simple car la ligne 4, si on avait employé des guillemets simples, aurait affiché littéralement `$mon_nom Rabelais`, c'est-à-dire que la variable n'aurait pas été extrapolée mais prise pour un mot commençant par le signe dollar.

Nous approfondirons dans le chapitre 3 les différences entre les guillemets simples et les guillemets doubles.

Les commentaires

Les commentaires sont uniquement destinés au programmeur. Ils ne seront ni affichés, ni exécutés. Ils permettent d'identifier telle partie de programme sans interférer dans le programme.

En HTML, le commentaire s'écrit ainsi `<!-- Ceci est un commentaire -->`.

En langage PHP, votre commentaire tient sur une ligne :

```
// Ceci est un commentaire sur mon code PHP
```

Mais si vous voulez documenter soigneusement votre programme (recommandé), vous avez toute licence d'écrire un commentaire de plusieurs lignes :

```
/* Ceci est un commentaire sur mon programme en PHP qui va me
permettre de retrouver mon aiguille dans le tas de foin qu'est
devenu mon code après moults ajouts */
```

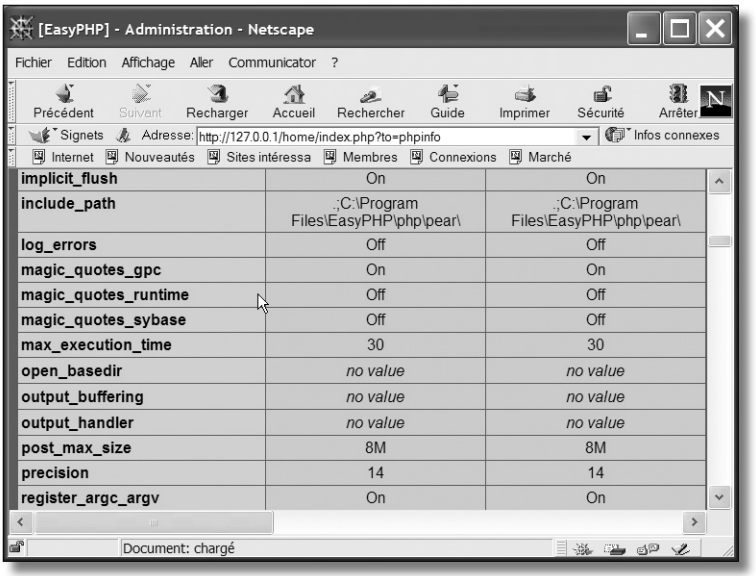
PHPinfo()

Sur votre machine, à la page d'accueil d'EasyPHP, cliquez sur le bouton **infos php**. Vous verrez apparaître la configuration de votre moteur PHP.

Vous pouvez visualiser cette liste de configuration sur le serveur de votre hébergeur en créant un fichier :

```
<?
phpinfo();
?>
```

Détruisez-le, après l'avoir sauvegardé sur votre machine, pour des raisons de sécurité. Il n'est pas bon de laisser des informations concernant les coulisses de votre site chez votre hébergeur, à la disposition du public. Il en sera de même pour un fichier qui contient un programme de courrier en nombre à partir de votre base.



implicit_flush	On	On
include_path	.;C:\Program Files\EasyPHP\php\pear\	.;C:\Program Files\EasyPHP\php\pear\
log_errors	Off	Off
magic_quotes_gpc	On	On
magic_quotes_runtime	Off	Off
magic_quotes_sybase	Off	Off
max_execution_time	30	30
open_basedir	*no value*	*no value*
output_buffering	*no value*	*no value*
output_handler	*no value*	*no value*
post_max_size	8M	8M
precision	14	14
register_argc_argv	On	On

▲ Figure 2.9 : *affichage de la fonction phpinfo()*

3

Les clefs du PHP

Dans ce chapitre, nous étudierons les règles de la syntaxe. Nous aborderons surtout la nature des variables et des valeurs, les opérateurs qui leur permettent d'interagir et les fonctions qui, à la manière de verbes, articulent les mots et les modifient. À la différence d'une langue, l'ordinateur compte et vérifie les valeurs, ce qui autorise les boucles et les tests. Nous œuvrons au cœur d'un langage très démocratique où l'information et le pouvoir sont largement distribués. Commençons par décliner les différents types de variable.

3.1 Variables scalaires

Les variables scalaires contiennent une seule valeur, par opposition aux tableaux qui en contiennent plusieurs.

Le nom de variable

Un nom de variable qui représente une valeur ou un ensemble de valeurs doit commencer par une lettre ou un blanc souligné (_), suivi de lettres, chiffres ou soulignés. Attention, le langage PHP distingue la casse dans le nom des variables. Ainsi `$Fromage` et `$fromage` sont des variables différentes.

Choix du nom des variables

Soyez cohérent dans le choix du nom des variables. L'objectif est d'obtenir un nom signifiant pour vous mais pas trop explicite. `$nbre_bouteille` sera aussi significatif que `$nombre_de_bouteilles` mais vous aurez moins de caractères à taper à chaque fois que vous l'utiliserez. Soyez rigoureux dans vos critères de nommage. Évitez les capitales car vous risquez de les oublier la deuxième fois que vous taperez le nom de la variable et d'engendrer ainsi une erreur avec une variable inexistante. Notez que le nom du champ d'un formulaire est utilisé pour nommer la variable qui en est issue, soyez rigoureux aussi sur ces noms de champ.

La valeur des variables scalaires peut être de plusieurs types. Il est bon de connaître ces derniers même si en général, le langage PHP fait le travail à votre place dans la plus grande transparence.

Le type des variables

Tab. 3.1 : Les différents types de variables

Type	En français	Exemple
`integer` ou `int`	Entier	45
`float`	Nombre à virgule	45.2
`string`	Chaîne de caractères	"J'veux faire du PHP!"
`boolean` ou `bool`	Booléen	`false` ou `true`

Les variables en langage PHP sont typées implicitement. Ainsi, PHP reconnaîtra un chiffre ou des lettres et le point du nombre à virgule flottante. Il vous laisse cependant la possibilité de forcer le type :

```
$code_postal="06400";
$code_postal= (int)$code_postal;
```

Ici, `$code_postal` qui serait naturellement une chaîne de caractères à cause des guillemets, peut être transtypé en nombre. Le forçage du type en `int` lui enlève le 0 du début. Par contre, si 06400 avait été sans guillemets, le nombre aurait interprété comme un octal à cause du 0 le précédant et aurait donc valu donc 3328 sur la deuxième ligne. Vous le verrez avec l'associativité des opérateurs, le sens du traitement avec l'affectation et le transtypage est de la droite vers la gauche. La variable `$code_postal` est ici transtypée puis affectée à elle-même, elle n'a plus rien d'un code postal. Faites bien la différence entre un chiffre et un nombre sinon, vous aurez des surprises.

Les entiers

Les variables de type entier sont assez simples à comprendre.

Vous pouvez écrire par exemple :

```
$a=1;
$b=2;
$c=$a+$b;
```

En l'occurrence, `$c` contient la valeur 3.

Si vous mettez les valeurs 1 et 2 entre guillemets doubles ou simples, PHP devine qu'il s'agit de valeurs numériques et non de chaîne de caractères grâce à l'opérateur +. Il fera, malgré tout, la somme des deux variables :

```
$a='1'; $b="2"; $c=$a+$b;
```

PHP vous simplifie la vie car il assume ces tâches de typage des données qui alourdissent inutilement le travail du programmeur et peuvent constituer une source d'erreurs.

Les nombres à virgule flottante

Méfiez-vous des `float`, c'est-à-dire des nombres à virgule flottante, car ils opèrent une approximation sur le résultat. Si vous avez comme résultat d'une opération un nombre tel que 6.555973333, il arrive souvent que les chiffres suivent la virgule à l'infini. Comme toute bonne chose a une fin, l'ordinateur opère une approximation sur le nombre pour ne pas générer un bogue de programme sans fin.

En anglais, le point remplace la virgule du français. Le type `float` comprend donc un point.

Nous verrons dans le chapitre **Des algorithmes, des outils et des fonctions** qu'il suffit de formater le nombre pour faire apparaître une virgule à la place du point.

Les booléens

Pour spécifier un booléen, vous pouvez utiliser `false` ou `TRUE`. Ce dernier est insensible à la casse.

Les valeurs considérées comme `false` sont :

- le booléen `FALSE` ;
- l'entier `0` (zéro) ;
- le nombre à virgule flottante `0.0` (zéro) ;
- la chaîne de caractères vide et le caractère `0` ;
- un tableau sans élément ;
- le type `NULL` (y compris les variables passées par la fonction `unset()`).

Les autres valeurs sont considérées comme `true`.

L'intérêt de cette liste est que les valeurs `false` répondront `true` au test `if(!$variable)`que nous évoquerons plus loin dans les structures de contrôle.

Pour l'inverse, `if ($variable)` est `true` si la variable existe et qu'elle est différente de `0`, de `NULL` ou de `""`. Par contre, si elle vaut `" "`, l'expression renvoie aussi la valeur `true`.

Les chaînes de caractères

Le type `string` est de loin le plus utilisé. Les espaces sont significatifs à l'intérieur d'une chaîne de caractères. Elle est encadrée par des guillemets simples (apostrophes) ou doubles (guillemets anglais) qui conditionnent l'interprétation des caractères de contrôle (nouvelle ligne, tabulation…) et des variables insérées.

Les guillemets

Il en existe deux types, sans compter les guillemets arrière (′), utilisés pour des commandes système sous Unix que nous n'évoquerons pas dans cet ouvrage.

Les guillemets simples

Vous l'avez vu dans le chapitre précédent, les chaînes de caractères insérées dans des guillemets simples sont affichées littéralement. Vous ne pouvez donc y utiliser des variables.

```
$une_phrase='je suis sur la page';
```

affichera la phrase intégrale. Comment faire pour une phrase contenant une apostrophe ?

```
$une_phrase='je m'y emploie';
```

Avec cette phrase, vous afficherez une erreur sur la page web. Pour y remédier, il vous suffit d'ajouter un signe d'échappement avant l'apostrophe.

Maintenant, vous obtenez :

```
$une_phrase='je m\'y emploie';
```

Cette phrase est parfaitement acceptée par le moteur PHP qui, s'il voit un guillemet simple commencer une chaîne, cherche le deuxième guillemet qui la clôt. Quand il trouve l'apostrophe, il conclut qu'elle sert de clôture à la chaîne, alors aidez-le avec une barre oblique inversée pour qu'il fasse la différence entre une apostrophe et un guillemet simple fermant.

Les guillemets simples sont très utiles au programmeur qui doit écrire un certain nombre de lignes HTML avec des `echo`. Il n'aura pas besoin de placer des caractères d'échappement devant chaque guillemet double délimitant les valeurs d'attribut HTML.

```
echo '<td bgcolor="#FFFFFF"><font face="Arial">Email:</font></td>';
```

sera correct. Vous disposez d'un outil très accommodant.

Les caractères d'échappement

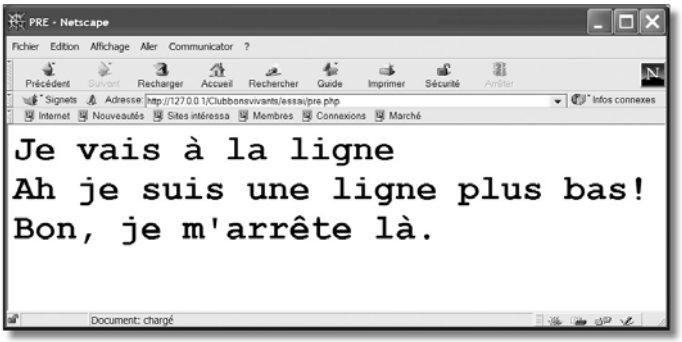
Le caractère d'échappement (ici, la barre oblique inversée) permet de signifier que le caractère suivant échappe à son interprétation normale.

Si vous allez à la ligne dans une chaîne entre guillemets simples, cette dernière affichera ce passage à la ligne :

Listing 3.1 : *<pre> avec des guillemets simples : pre.php*

```
<?
$phrase='<PRE>Je vais à la ligne
Ah je suis une ligne plus bas!
Bon, je m\'arrête là.</PRE>';
echo "<font size=\"7\"><b>$phrase</b></font>";
?>
```

La page web affichera le passage à la ligne.



▲ Figure 3.1 : *Passage à la ligne à l'intérieur de guillemets simples*

La balise <PRE>

La balise HTML `<pre>` demande au navigateur d'afficher la page avec les attributs littéraux du texte en police `courrier`. Pour afficher la ligne, il créera une balise `<br>` à chaque retour chariot.

Les guillemets doubles

Les guillemets doubles servent aussi à caractériser des chaînes de caractères, avec cependant quelques nuances.

Comme nous l'avons vu, les doubles guillemets font une extrapolation de variable, c'est-à-dire qu'ils remplacent le nom des variables par leur valeur. D'autre part, ils permettent d'utiliser des syntaxes spéciales d'échappement.

Prenons un exemple tiré du poème l'Âme du vin de Charles Baudelaire :

Listing 3.2 : *L'Âme du vin : ame_vin.php*

```
$vers1="Un soir, l'âme du vin chantait dans les bouteilles!<BR>\n";
$vers2=" – \"Homme, vers toi je pousse, ô cher déshérité,<BR>\n";
$vers3="Sous ma prison de verre et mes cires vermeilles,<BR>\n";
$vers4="Un chant plein de lumière et de fraternité!<BR>\n";
$strophe=$vers1.$vers2.$vers3.$vers4;
```

Cette strophe demande quelques éclaircissements.

Chaque ligne est close par une balise <BR> et une séquence d'échappement \n. Nous avons vu que la balise <BR> affichait une fin de ligne dans une page web HTML. Le \n aussi. Le \n, exclusivement dans des guillemets doubles, constitue une fin de ligne. Pourquoi mettre deux fois une fin de ligne ? Simplement pour obtenir un code source propre et lisible. La source HTML renvoyée par le moteur PHP sera :

```
Un soir, l'âme du vin chantait dans les bouteilles!<BR>
— "Homme, vers toi je pousse, ô cher déshérité,<BR>
Sous ma prison de verre et mes cires vermeilles,<BR>
Un chant plein de lumière et de fraternité!<BR>
```

La séquence d'échappement \n n'est pas répercutée dans l'affichage de la page mais dans l'affichage du code source HTML.

D'autre part, vous avez remarqué sans doute que la séquence \" permettait d'insérer des guillemets doubles dans une chaîne encadrée elle-même de guillemets doubles, comme pour l'apostrophe précédemment. Par contre, l'apostrophe n'a pas besoin de barre oblique inversée à l'intérieur de doubles guillemets, c'est logique.

Tab. 3.2 : Les séquences d'échappement

Séquence d'échappement	Définition
\n	Nouvelle ligne
\r	Retour chariot (utilisé uniquement sous Windows)
\t	Tabulation
\$	Le signe Dollar
\\	Barre oblique inversée

Les constantes

Les constantes représentent un confort pour le programmeur. Elles évitent une perte de temps et les programmes gagnent en lisibilité. Elles s'écrivent en capitales.

Pour les déclarer, utilisez la fonction `define()` :

```
<?
define ('PI', 3.1415);
define ('MAIL', 'LivrePHP@carfantan.com');
?>
```

Il est plus facile de taper `PI` que `3.1415` et si vous décidez d'ajouter des décimales, il suffit de le faire une fois pour chaque fichier (`3.1415926535`). La plupart des logiciels permettent un remplacement approfondi et automatique dans tous les fichiers à l'intérieur d'un répertoire. Ici, il sera plus aisé de changer uniquement cette ligne dans chaque fichier.

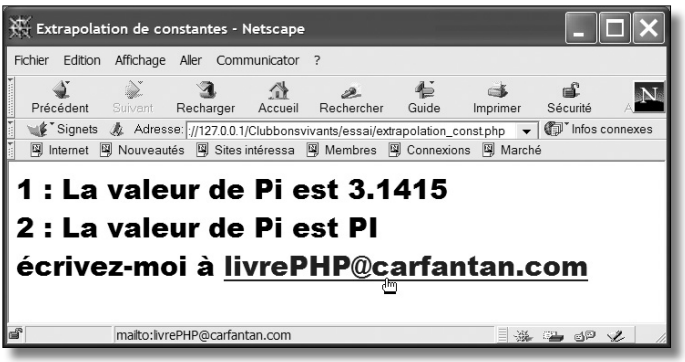
Extrapolation des constantes

Les constantes ne sont pas extrapolées à l'intérieur d'une chaîne de caractères.

Listing 3.3 : *Extrapolation d'une constante : extrapolation_const.php*

```
<?
echo '<font face="Arial Black" size="5">';
echo "1 : La valeur de Pi est ".PI.'<BR>';
echo "2 : La valeur de Pi est PI<BR>";
echo "écrivez-moi à <A HREF=\"mailto:".MAIL."\">".MAIL."</A>";
echo '</font>';
?>
```

La constante n'est pas à l'intérieur des guillemets. Elle est concaténée entre les deux parties de la phrase. Dans la ligne numérotée 2, elle n'est pas extrapolée, étant à l'intérieur des parenthèses.



▲ Figure 3.2 : *Extrapolation d'une constante*

Extrapolation de variable

Il arrive que vous soyez obligé de coller une variable à des caractères. S'il s'agit de deux variables, cela fonctionne à condition de les encadrer de guillemets doubles :

```
$annonce="Félicitations, vous y êtes arrivé";
$genre='e';
$fin_annonce=", vous gagnez une bouteille de Bourgueil!";
echo $annonce$genre$fin_annonce;
```

Le résultat affichera : `Félicitations, vous y êtes arrivé, vous gagnez une bouteille de Bourgueil!`.

Par contre, si vous désirez coller une variable devant un mot quelconque, le moteur PHP prendra les deux mots pour une seule variable. Vous pouvez malgré tout l'afficher en utilisant des accolades :

```
$nombre=10;
echo "Patience! c'est la ${nombre}e fois que vous cliquez pour
envoyer ce formulaire que nous sommes déjà en train de traiter";
```

Le résultat affichera : `Patience! c'est la 10e fois que vous cliquez pour envoyer ce formulaire que nous sommes déjà en train de traiter`.

Le langage PHP a été forgé par des utilisateurs confrontés à tous les problèmes possibles sur Internet. Des solutions adaptées ont été apportées au fil des versions. Seule le Libre pouvait le permettre.

Construire une variable dynamique

Le langage PHP est un langage de bricoleurs mais on ne bricole pas sans une certaine rigueur. Ainsi, il est possible de construire une variable à partir d'une autre variable qui fournit un nom en dynamique :

Listing 3.4 : *Variables dynamiques : var_dynamique.php*

```
<?
// commencez par créer la variable qui donnera le nom
$nom_au_vol="nom_doiseau";
// ensuite donnez une valeur à la variable dynamique
$$nom_au_vol='Pigeon';
//enfin vous pouvez l'utiliser
```

```
echo $nom_doiseau;
//ici le programme affichera Pigeon
?>
```

Nous en verrons les applications dans le chapitre **Des algorithmes, des outils et des fonctions**.

Les tableaux

Le tableau est une variable qui contient plusieurs valeurs. Il existe deux types de tableaux : les tableaux indexés par des nombres de 0 à n et les tableaux indexés par des noms.

Les noms de tableaux

Réservez le pluriel dans les noms de variable aux tableaux car en PHP, rien ne distingue le nom d'un tableau du nom d'une variable scalaire.

L'index est le groupe d'indices qui permet de classer les données à l'intérieur d'un tableau afin de pouvoir insérer des valeurs et les retirer le cas échéant.

Les tableaux à index de noms sont des tableaux de hachage et ceux à index numérique sont appelés simplement des tableaux. Enfin, les tableaux multidimensionnels sont des tableaux qui contiennent d'autres tableaux et non des valeurs scalaires.

Les tableaux à index numérique

Pour entrer des données dans un tableau, il suffit d'utiliser les crochets :

```
$mon_tableau[]=$ma_case1;
```

La valeur de `$ma_case1` se logera dans la première cellule du tableau, qui est à l'indice 0. La deuxième valeur sera logée dans la cellule d'indice 1, etc.

Vous pouvez choisir la case où sera insérée votre valeur :

```
$mon_tableau[4]=$ma_case1;
```

Ici, la valeur sera insérée dans la 5e case.

Tout tableau commence à l'indice 0

La première case étant la case 0, ne confondez pas la place de la dernière cellule dans le tableau et le nombre de cellules dans le tableau. Si vous avez cinq cases, l'index de la dernière est 4.

Pour récupérer une valeur du tableau, il suffit de connaître sa place :

```
$ma_valeur=$mon_tableau[4];
echo $ma_valeur;
echo "<br>";
//une autre façon de faire
echo "$mon_tableau[4]";
```

Les tableaux de hachage (ou tables d'adressage associatif)

Ces tableaux sont indexés par nom, ce qui est souvent plus pratique, un peu comme les URL d'Internet.

Il est plus facile de taper et de mémoriser `www.multidimensionalmusic.com` que **207.158.192.40**, numéro IP du serveur.

Nous utilisons le même système pour insérer une donnée ou la récupérer mais il est plus facile de retrouver une donnée avec un nom.

Les guillemets simples pour l'indice nominal

N'oubliez pas les guillemets simples pour encadrer le nom à l'intérieur des crochets :

```
$mon_tableau_hachage['François Rabelais']='Gargantua';
$livre=$mon_tableau_hachage['François Rabelais'];
echo $livre;
//affichera Gargantua
//une autre façon de faire
echo $mon_tableau_hachage['François Rabelais'];
```

Si pour la dernière ligne de l'exemple, vous utilisez des guillemets avec `echo`, vous afficherez une erreur :

```
$mon_tableau_hachage['François Rabelais']='Gargantua';
echo "$mon_tableau_hachage['François Rabelais']";
//ce n'est pas valide
echo "{$mon_tableau_hachage['François Rabelais']}";
/*ici grâce aux accolades, la page affichera la
valeur correspondant à François Rabelais :
Gargantua.*/
```

Le nom ou le nombre qui indexe est appelé une clé et la valeur correspondante une valeur. Pour ce type de tableau, on parle d'une paire clé/valeur.

Clé unique

Chaque clé doit être unique sous peine d'effacer la valeur précédente de clé identique.

Les tableaux multidimensionnels

Ce type de tableau est formé de tableaux de hachage ou de tableaux, ou d'un mélange des deux.

La clé s'appuie sur le même principe que les deux types de tableaux précédents mais la valeur peut être un tableau à 1 ou n tranches.

Rabelais a écrit plusieurs livres. Créons un nouveau tableau de hachage contenant un tableau :

```
$livres[]="Gargantua";
$livres[]="Pantagruel";
$auteurs ['François Rabelais']=$livres;
echo $auteurs ['François Rabelais'][1];
//affichera Pantagruel
```

Essayons maintenant un tableau de hachage contenant un tableau de hachage :

```
$livres['premier livre']="Gargantua";
$livres['deuxième livre']="Pantagruel";
$auteurs ['François Rabelais']=$livres;
echo $auteurs ['François Rabelais']['premier livre'];
//affichera Gargantua
```

Nous n'avons pas encore essayé un tableau contenant un tableau de hachage :

```
$livres['premier livre']="Gargantua";
$livres['deuxième livre']="Pantagruel";
$auteurs []=$livres;
echo $auteurs [0]['premier livre'];
//affichera Gargantua
```

Enfin, il n'y a pas de raison de délaisser le tableau contenant un tableau, surtout qu'il se révèle très utile dans les boucles, comme nous le verrons dans le chapitre *MySQL en action* :

```
$livres[]="Gargantua";
$livres[]="Pantagruel";
$auteurs []=$livres;
echo $auteurs [0][1];
//affichera Pantagruel
```

Vous pouvez mélanger les deux types d'index dans le même tableau :

```
$livres[6]="Pantagruel";
$livres['premier livre']="Gargantua";
$livres[]="Tiers livre";
/* la clef sera donc 7, un numéro au-dessus de 6 l'indice maximum
précédent*/
$livres[6]="Quart livre";
/* cette dernière insertion écrasera l'ouvrage Pantagruel qui a le
même indice*/
$auteurs []=$livres;
echo $auteurs [0][6];
//affichera Quart livre
```

Vous pouvez aussi empiler des tableaux sur autant de niveaux que vous pouvez en concevoir mais vous aurez peu d'occasions de les utiliser sur autant de niveaux et perdriez beaucoup de temps à les concevoir.

3.2 Les opérateurs

Les opérateurs fournissent les fonctions essentielles à un langage de programmation pour que les variables interagissent.

Les opérateurs arithmétiques

Tab. 3.3 : Les opérateurs arithmétiques

Nom	Exemple	Action
Addition	`$a+$b`	Additionne les valeurs de `$a` et de `$b`.
Soustraction	`$a-$b`	Soustrait la valeur de `$b` de la valeur de `$a`.
Multiplication	`$a*$b`	Multiplie les valeur de `$a` et de `$b`.
Division	`$a/$b`	Divise la valeur de `$a` par la valeur de `$b`.
Modulo	`$a%$b`	Retourne le reste de la division de la valeur de `$a` par la valeur de `$b`.

Les opérateurs d'affectation

L'opérateur d'affectation est le signe `=`. Ne le confondez pas avec l'opérateur de comparaison `==` (égal à).

Ainsi, dans `$a=3`, la variable `$a` reçoit la valeur 3.

Des déclinaisons de cet opérateur proposent au programmeur des raccourcis. Il suffit de combiner le signe égal à un opérateur arithmétique, à un opérateur de bit ou à l'opérateur de concaténation :

```
$a=6;
$a+=3; // $a prend la valeur   de 9
$a-=2; // $a prend la valeur   de 7
$a%=3; //$a prend la valeur de 1 (reste de la division de 7/3)
$a="Bonjour";
$a.=", il fait beau"; // résultat : Bonjour, il fait beau
```

Les opérateurs de bits

Chaque caractère est composé d'un octet, c'est-à-dire 8 bits (0 ou 1). Ainsi, la lettre A est codée 01000001 et la lettre a, 01100001.

Les opérateurs portent surtout sur les bits positionnés sur la valeur 1.

Tab. 3.4 : Les opérateurs de bits

Exemple	Nom	Effet
`$a & $b`	AND (et)	Les bits à 1 dans les deux variables restent à 1.
`$a | $b`	OR (ou)	Les bits à 1 dans l'une ou les deux variables restent à 1.
`$a ^ $b`	XOR (ou exclusif)	Chaque bit à 1 dans $a ou dans $b reste à 1 sauf si le bit est à 1 pour les deux variables. Dans ce cas, il est positionné à 0.
`~ $a`	NO (non)	Chaque bit est inversé. Le bit à 1 est positionné à 0 et vice versa.
`$a<< $b`	Décalage à gauche	Chaque bit est décalé de $b positions vers la gauche. Les bits de gauche se retrouvent à droite dans le même ordre.
`$a>> $b`	Décalage à droite	Chaque bit est décalé de $b positions vers la droite. Le 8e bit devient le premier, etc.

Ces opérateurs servent à crypter un texte, par exemple. En décalant les bits vers la gauche ou la droite avec une clé de 1 à 8, vous modifiez les caractères d'un texte. Il suffit ensuite de faire l'opération inverse pour avoir le texte en clair. C'est comme un cercle où les deux extrémités se rejoignent et où les 0 et les 1 se décalent dans un sens ou dans l'autre. Cette méthode se combine avec des algorithmes plus complexes.

Les opérateurs de comparaison

Ces opérateurs sont très souvent utilisés pour tester des valeurs avec un `if`.

Tab. 3.5 : Les opérateurs de comparaison

Exemple	Nom	Résultat
`$a== $b`	Égal	Vrai si `$a` est égal à `$b`
`$a=== $b`	Identique	Vrai si `$a` est égal à `$b` et de même type
`$a!= $b` ou `$a<> $b`	Différent	Vrai si `$a` est différent de `$b`
`$a!== $b`	Non identique	Vrai si `$a` est différent de `$b` ou s'ils ne sont pas de même type
`$a < $b`	Plus petit	Vrai si `$a` est plus strictement petit que `$b`
`$a > $b`	Plus grand	Vrai si `$a` est strictement plus grand que `$b`
`$a <= $b`	Inférieur ou égal	Vrai si `$a` est plus petit ou égal à `$b`
`$a >= $b`	Supérieur ou égal	Vrai si `$a` est plus grand ou égal à `$b`

Il reste un opérateur de comparaison que nous présentons à part, l'opérateur ternaire ?.

```
$a==$b ? $c=8 : $c=2;
```

Si la valeur de `$a` est égale à la valeur de `$b`, `$c` prend la valeur 8, sinon `$c` prendra la valeur 2.

On l'utilise surtout pour des expressions courtes. On peut remplacer l'exemple par :

```
if ($a==$b){
$c=8;
}
else {
$c=2;
}
```

Nous approfondirons les tests et les boucles plus loin dans ce chapitre.

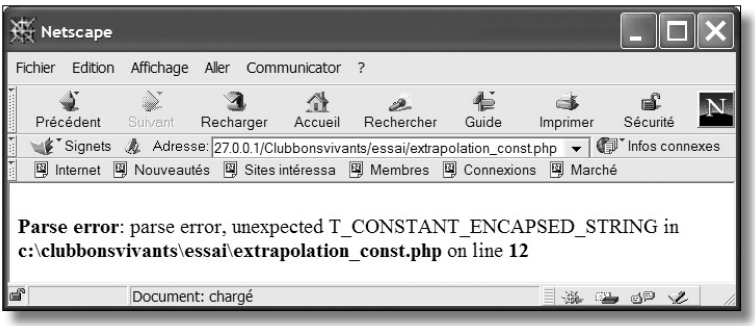
Les opérateurs de contrôle d'erreur

En PHP, les erreurs sont signalées sur la page directement avec le nom du fichier et le numéro de ligne d'où l'erreur provient. Nous verrons dans le chapitre *Mettre de l'ordre* que le numéro de la ligne ou le nom du fichier peuvent être différents de celui qui est indiqué pour des raisons que nous éclaircirons, mais en général il est assez précis. Par contre, nous n'avons pas forcément intérêt à ce qu'un visiteur y accède. C'est pourquoi avant de mettre en ligne vos pages, vous ferez des tests approfondis puis utiliserez l'opérateur `@`. Il est collé devant une fonction.

Nous l'utilisons surtout pour les fonctions de connexion avec les bases de données. Voici par exemple une fonction que nous verrons dans le chapitre suivant :

```
$resultat=@mysql_db_query($dbname,$sql,$id_link);
```

Sans cet opérateur, il suffit que la base soit momentanément indisponible pour obtenir des messages disgracieux.



▲ Figure 3.3 : *Affichage d'une erreur*

Il vaut mieux contrôler l'erreur et écrire un message personnalisé. Nous verrons cela dans le chapitre *Mettre de l'ordre*.

Aucune partie du programme ne sera exécutée si une erreur est décelée par le compilateur. C'est un avantage car un programme tronqué peut être dangereux et générer par exemple des boucles infinies. Il faut préciser que

PHP possède des garde-fous. Si le programme met plus de 30 secondes pour s'exécuter, il est arrêté. Le temps maximum d'exécution est indiqué à la rubrique *max_execution_time* dans le fichier créé par la commande `phpinfo()`. Attention toutefois de n'utiliser cet opérateur qu'une fois vos programmes finalisés. Autrement, certaines parties pourraient ne pas s'exécuter, à votre insu.

Les opérateurs unaires

PHP permet d'incrémenter ou de décrémenter une variable numérique ou une chaîne de caractères.

Il s'écrit `++$a`, `$a++` ou `--$a` et `$a--`.

Les variables affublées de `++` s'incrémentent elles-mêmes de 1 tandis que celles sur lesquelles est appliqué `--` se décrémentent de 1.

Selon que les opérateurs sont placés avant ou après, la variable prend la nouvelle valeur avant ou après. Voici un exemple :

```
$a=2;
echo ++$a; // affichera 3
//maintenant $a vaut 3;
echo $a; // affichera 3
echo $a++; // affichera 3
echo $a; // affichera 4
```

L'opérateur `--` se comporte de la même façon.

En ce qui concerne les caractères, vous pouvez utiliser l'opérateur unaire mais seulement pour incrémenter. C'est parfois très utile. Imaginez que vous vouliez une clé unique de trois caractères pour une table qui comprend de nombreuses entrées. Vous pouvez le faire à la main ou utiliser les lettres avec une boucle et un opérateur unaire. Nous le verrons dans le chapitre *Des algorithmes, des outils et des fonctions*. Sur deux caractères, vous avez 674 possibilités et sur trois, 17574 combinaisons possibles. Pour les chiffres, vous avez respectivement 99 et 999 combinaisons.

Les opérateurs logiques

Tab. 3.6 : Les opérateurs logiques

Exemple	Nom	Résultat
`$a and $b ou $a && $b`	AND (et)	Vrai si `$a` et `$b` sont vrais
`$a or $b ou $a || $b`	OR (ou)	Vrai si `$a` ou `$b`, ou les deux, sont vrais
`$a xor $b`	XOR (ou exclusif)	Vrai si `$a` ou `$b` sont vrais, mais pas les deux ensemble
`!$a`	NOT (non)	Vrai si `$a` est faux

Attention, `&&` et `and` n'ont pas la même précédence, de même pour `||` et `or`, comme nous le voyons ci-dessous.

Priorité des opérateurs

La priorité spécifie la force du lien entre deux membres associés par un opérateur. Prenons un exemple simple dans les opérateurs arithmétiques. Le résultat de `8+7*10` sera `78` car la multiplication a priorité sur l'addition, donc le moteur PHP commencera par multiplier `7` par `10` avant d'additionner le résultat avec `8`.

La façon de forcer la priorité et donc de résoudre ce problème est d'ajouter des parenthèses. Le résultat de `(8+7)*10` sera `150`. Attention de ne pas systématiquement éluder le problème en ajoutant des parenthèses inutiles. Si c'est moins élégant, les collections de parenthèses peuvent également devenir illisibles. En cas de doute, la parenthèse est plus simple mais d'un autre côté, il vaut mieux s'habituer à écrire un code élégant et à tester ses programmes pour détecter les erreurs. Une fois les nuances assimilées, vous gagnerez en temps de codage et en lisibilité et aurez la satisfaction d'avoir un code élégant et maîtrisé.

Voici un tableau représentant les opérateurs dans un ordre de priorité croissante.

Tab. 3.7 : Priorité des opérateurs

Associativité	Opérateur
Gauche	`or`
Gauche	`xor`
Gauche	`and`
Gauche	`print`
Droite	`= += -= *= /= .= %=`
Gauche	`? :`
Gauche	`||`
Gauche	`&&`
Gauche	`|`
Gauche	`^`
Gauche	`&`
Non associatif	`== != === !==`
Non associatif	`< <= > >=`
Gauche	`<< >>`
Gauche	`+ - .`
Gauche	`* / %`
Droite	`! ~ ++ -- (int) (double) (string) (array) @`

À priorité égale d'opérateurs, l'opération est choisie de la gauche vers la droite, ou vice versa, selon le côté indiqué.

Les opérateurs de chaîne de caractères

Il existe deux opérateurs `.` et `.=` que nous avons déjà évoqués dans le chapitre 2. Vous pouvez également les appliquer à des données numériques :

Listing 3.5 : *Opérateurs de chaînes de caractères : chaine_car.php*

```
<?
$premier=4;
$deux=2;
```

```
$type=gettype($deux);
echo "$deux : $type<br>";//affiche 2 un entier
$nombre=$premier.$deux;
$type=gettype($nombre);
echo "$nombre : $type<br>";//affiche 42 une chaîne
$nombre.=3;
$type=gettype($nombre);
echo "$nombre : $type<br>"; //affiche 423 une chaîne de caractères
$type=gettype($nombre);
$nombre+=$nombre;
$type=gettype($nombre);
echo "$nombre : $type<br>"; //affiche 846 un entier
?>
```

Avec PHP, les données se transforment de manière transparente. PHP a traité ces nombres comme des caractères qu'il a concaténés. Mais le résultat est-il un nombre ou une chaîne de caractère ? Ici, nous utilisons la fonction `gettype()` pour afficher le type de la donnée.

Les deux premières variables sont des entiers. Ensuite, la variable `nombre` devient une chaîne de caractères puis elle redevient un entier. Que de métamorphoses ! Le type des données peut varier, il dépend de l'opérateur. Pour une concaténation, les variables en présence deviennent des chaînes et dans le cas d'une addition, le chiffre n'est plus un caractère mais un nombre.

3.3 Les structures de contrôle

Les structures de contrôle sont caractérisées par des parenthèses parfois accompagnées d'accolades. Elles évaluent la validité d'une expression et décident de l'exécution ou non d'une partie du programme.

Les tests

Les tests sont déclenchés par l'expression `if` ou `switch`.

If

`if` est un aiguillage. Le moteur PHP, comme un train, suit la ligne du programme jusqu'à l'aiguillage qui teste sa variable et en fonction de sa réponse, le fait bifurquer dans telle ou telle direction.

Voici la forme commune d'un `if` :

If (expression vraie utilisant un opérateur logique) {

Exécution 1;

}

else {

Exécution 2 ;

}

`else` intervient dans tous les autres cas, c'est-à-dire quand l'expression est fausse. Si l'expression est vraie, le code inclus dans le bloc `else` matérialisé par les accolades n'est jamais exécuté. De la même façon, si l'expression est fausse, le code inclus dans le bloc `if` n'est pas exécuté. `else` n'est pas obligatoire. `if`, lorsqu'il est seul, se contente de conditionner une exécution de code. Voici deux exemples :

Listing 3.6 : *Combien y-a-t-il de minutes dans une heure ? : heure.php*

```
<?
$mn=5;
$mn*=13;
echo "1) nombre de minutes : $mn<br>";
if ($mn>60){
 $h=$mn/60;  // division par 60 pour avoir les heures
 $mn%=60;// modulo par 60 pour avoir les minutes restantes
 echo "cela fait ${h}h ${mn}mn";
 echo " (<i>c'est brut!)</i><br>"; //petit commentaire perfide!!!
 $h=round($h); //nous affinons l'affichage
 echo '<font color="#FF0000">$h=round($h);</font><br>';
 echo "cela fait ${h}h ${mn}mn";
 echo " <i>(un peu de finesse dans un monde de brutes!)</i><br>";
//la maxime de Superprogrammeur !!!
}
///////////////fin du if minutes >60/////////////////////
else {
 echo "cela fait ${mn}mn<br>";
}
```

Ce premier exemple est un aiguillage avec deux blocs initialisés par `if` et `else`. Il est uniquement destiné à votre apprentissage car un programmeur utiliserait plutôt les fonctions d'heure. Remarquez les retraits qui permet-

tent de mieux lire le code ainsi que les commentaires, car les accolades sont parfois éloignées et il est bon d'y ajouter un commentaire pour les préciser.

Dans le deuxième exemple, nous allons améliorer un code précédent :

```
$annonce="Félicitations, vous y êtes arrivé";
if ($sexe=='f'){
  $genre='e';
}
$fin_annonce=", vous gagnez une bouteille de Bourgueil";
echo $annonce$genre$fin_annonce;
```

Ici, nous n'avons pas besoin de `else` car si `$sexe=='h'`, la variable `$genre` ne sera pas initialisée et le `e` n'apparaîtra pas. Si la variable n'existe pas, PHP ne génère pas d'erreur, ce qui économise du code mais peut faire perdre du temps en cas d'erreur.

Test sur l'opérateur ==

Pour un test de validité, si vous utilisez le signe = au lieu du signe == à l'intérieur de l'expression conditionnelle, cela ne provoque aucune erreur mais le code sera toujours exécuté car vous assignez la valeur à la variable au lieu d'en tester la valeur.

`elseif` permet d'ajouter plusieurs possibilités d'aiguillage entre les blocs `if` et `else`. `else` est le bloc par défaut, il n'est jamais obligatoire :

```
<?
If ($age<18){
Bloc d'instructions
}
elseif ($age>=18 && $age<35){
Bloc d'instructions
}
elseif ($age>=35 && $age<49){
Bloc d'instructions
}
else {
Bloc d'instructions
}
?>
```

Switch

Maintenant, vous pouvez utiliser `switch` qui correspond à un bloc `if`, `elseif` mais avec des valeurs ponctuelles. Il ne s'applique pas à des valeurs sur des intervalles. Ici les valeurs (1, 2…) sont des codes donnés par convention, par exemple dans un menu déroulant à l'intérieur d'un formulaire, comme nous le verrons dans le prochain chapitre.

```
switch ($age){
case 1:
   $age="moins de 18 ans";
   break;
case 2:
   $age="entre 18 et 35 ans";
   break;
case 3:
   $age="entre 35 et 49 ans";
   break;
default:
   $age="plus de 49 ans";
}
```

Si `$age` est égal à 1, il prend la valeur de `moins de 18 ans`. Le `default` est la valeur prise par `$age` s'il n'est égal à aucune des valeurs proposées.

3.4 Les boucles

Qu'est-ce qui identifie un langage comme langage de programmation ? La boutade est de répondre que ce sont les boucles et nous ne sommes pas loin de la vérité. Vous êtes sûr de les utiliser dans vos programmes et bien souvent, elles seront au cœur de vos algorithmes.

Algorithme

Ensemble des règles opératoires propres à un calcul (extrait du dictionnaire Micro-Robert).

Les boucles permettent d'exécuter un bloc d'instructions autant de fois qu'il est nécessaire. On utilise un index initialisé qui s'incrémente à chaque boucle. Dans l'intitulé, on insère une condition de sortie de boucle.

La boucle while

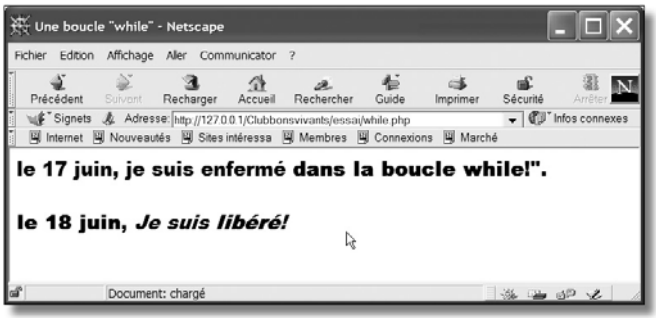
```
while   (expression est vrai){
Bloc instruction
}
```

Vous pouvez utiliser une expression déjà initialisée ou en créer une avec une nouvelle variable numérique (en général, on l'appelle i comme index). Quand vous utilisez un index, il faut obligatoirement trois composantes : la valeur initiale, la valeur limite à partir de laquelle la valeur n'est plus vraie et la valeur d'incrémentation ou de décrémentation.

Nous allons localiser la variable pour que vous saisissiez le moment où elle sort de la boucle. Prenons, par exemple, une personne condamnée à douze jours de prison. Elle rentre le 1er mai et sort le 12. Nous utiliserons une boucle `while`. Tant que la variable est dans la boucle, le prisonnier est incarcéré et la sortie de la boucle correspond à sa levée d'écrou :

Listing 3.7 : *Boucle while : while.php*

```
<?
$i=1;
$temps=12; //premier jour de tôle!!!
while ($i<$temps){
   echo "le $i";
   if ($i==1){
      echo "<sup>er</sup>";
/*cette balise représente la position supérieure en puissance,
  pour celle en indice qui est en une position inférieure,
  utilisez <sub></sub>*/
   }
   echo " mai, je suis <B>en prison</B>!\".<br>";
   //mettons le en gras
   $i++;
   // Oubliez cette ligne et vous générez une boucle infinie
}
echo "<br><b>le $i mai</b>, <i>Je suis <B>libéré</B>!</i>";
//enfin je suis libre!!!
?>
```



▲ Figure 3.4 : *Une boucle while*

La boucle do … while

Cette boucle est la même que la boucle `while` mais à la différence de `while`, elle initie un tour avant d'entrer dans la boucle, que la condition soit vérifiée ou non, car la valeur est testée en fin de boucle. On l'emploie peu mais elle est utile parfois pour décrémenter ou incrémenter une limite avant une boucle.

Notre prisonnier récidive. Il est condamné par le juge `do` … et gardé par le surveillant `while`. Ici, la boucle est parcourue au moins une fois sans que la condition ne soit testée :

Listing 3.8 : *Boucle do... while : do_while1.php*

```
<?
$i=1;
$temps=12;
do {
   echo "le $i";
   if ($i==1){
      echo "<sup>er</sup>";
   }
   echo " juin, je suis <B>en prison</B>!\".<br>";
   // c'est le mois suivant
   $i++;
   /*Si vous oubliez cette ligne, vous générez une
   boucle infinie */
}
while ($i<$temps);
echo "<br><b>le $i juin</b>, <i>Je suis <B>libéré</B>!.</i>";
?>
```

Il aurait dû subir sa peine mais imaginons que le voleur ait indemnisé la victime, il entame un recours mais le temps que la justice tranche... Nous initialisons la variable à 17, le jour de son entrée :

Listing 3.9 : *Boucle do... while : do_while2.php*

```
<?
$i=17; //il entre le 17 juin en prison
$temps=0;
do {
   echo "le $i";
   if ($i==1){
      echo "<sup>er</sup>";
   }
   echo " juin, je suis <B>en prison</B>!\".<br>";
   // c'est le mois suivant
   $i++;
   //Si vous oubliez cette ligne, vous générez une boucle infinie
}
while ($i<$temps);
echo "<br><b>le $i juin</b>, <i>Je suis <B>libéré</B>!.</i>";
?>
```

Le point virgule dans le do... while

N'oubliez pas le point-virgule après la parenthèse du `while` dans la boucle `do … while`.

Il passe au moins un jour en prison avec le juge `do …` et le surveillant `while`, alors qu'il n'en passerait aucun avec le juge `while`.

En général, avec une valeur d'index, il est préférable d'utiliser la boucle `for`, faite sur mesure, comme nous le voyons ci-dessous.

La boucle for

`for` utilise un compteur qu'elle initialise, limite et incrémente afin d'exécuter son bloc d'instructions. Elle concentre les trois lignes séparées par des points-virgules à l'intérieur de sa parenthèse :

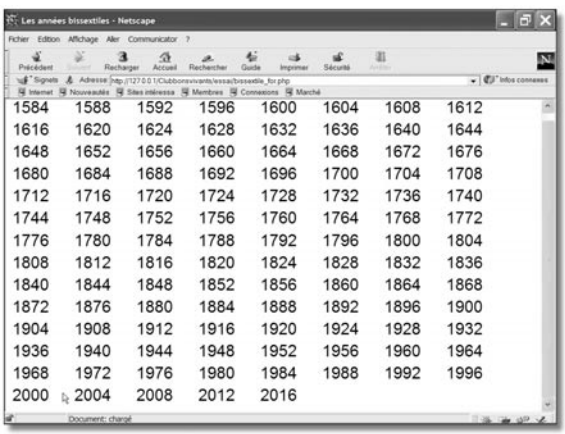
```
for (initialisation ; limite ; incrémentation){
Bloc d'instructions
}
```

Notre calendrier bancal va vous permettre de comprendre les possibilités de cette boucle `for`. Nous allons afficher les années bissextiles qui ont pour particularité de contenir 366 jours. Le calendrier occidental actuel, le calendrier grégorien, a été institué en 1582 par le pape Grégoire. Pour l'heure d'été, nous verrons plus loin dans le chapitre. Une année est bissextile si elle est divisible par 4. Les années de siècle comme 1900 ou 2000 sont bissextiles si elles sont divisibles par 400. Nous allons utiliser l'opérateur % (modulo) pour vérifier si l'année est divisible par 4 et le siècle par 400.

Comme vous le voyez, nous initialisons le calendrier à l'an 1582 :

Listing 3.10 : *Les années bissextiles1 : bissextile1.php*

```
<?
for ($an=1582;$an<2020;$an++){
   if ($an%100==0){
      if ($an%400==0){
         echo "$an<br>";
      }
   }
   elseif ($an%4==0){
      echo "$an<br>";
   }
}
?>
```



▲ Figure 3.5 : *Les années bissextiles*

Le calendrier maya est plus régulier et plus précis mais il ne nous aurait pas permis de vous montrer toutes ces subtilités, comme il n'aurait pas permis à Michèle Morgan, née un 29 février, de dire qu'elle vieillit quatre fois moins vite que la normale.

Il est possible d'ajouter un autre indice dans la boucle `for`. Ainsi, si nous voulons donner le nombre de jours depuis l'an 1582, ajoutons une variable `$j` avec une virgule. Elle représente le nombre de jours à partir du premier janvier 1583 :

Listing 3.11 : *Les années bissextiles2 : bissextile2.php*

```
for ($an=1582,$j=0;$an<2020;$an++,$j=$j+365){
   // remarquez les virgules
   //vous pouvez aussi mettre $j+=365 à la place de $j=$j+365
   if ($an%100==0){
      if ($an%400==0){
         $j++;
         echo "$an : $j jours<br>\n";
      }
   }
   elseif ($an%4==0){
      $j++;
      echo "$an : $j jours<br>\n";
   }
}
?>
```

Foreach

La boucle `foreach` n'existe qu'à partir de PHP4. Dans PHP3, vous utiliserez les fonctions `each` … `list`. `foreach` est utilisé pour parcourir les tableaux et en extraire les clefs et les valeurs. Pour vous montrer un exemple, nous prendrons les dix premiers départements français :

```
$departements=array('01' => 'Ain', '02' => 'Aisne', '03' => 'Allier',
'04' => 'Alpes de Haute Provence', '05' => 'Hautes Alpes', '06' =>
'Alpes Maritimes', '07' => 'Ardèche', '08' => 'Ardennes', '09' =>
'Ariège', '10' => 'Aube');
```

Voici donc un nouvel opérateur pour créer un tableau `=>`. Cet opérateur est spécifique aux tableaux et lie la clé à sa valeur. Notez que '01' n'est pas un indice numérique mais une chaîne de caractères. Si nous avions écrit '1', cette clé aurait été interprétée comme un indice numérique par PHP.

Maintenant, nous allons l'afficher :

Listing 3.12 : *Les départements : departements.php*

```
foreach($departements as $clef=>$valeur){
   echo "$clef => $valeur<br>\n";
}
```

Si vous ne voulez afficher que la valeur :

```
foreach($departements as $valeur){
   echo " $valeur<br>\n";
//Nous avons ajouté \n pour que le code source soye plus beau!
}
```

each list

Voici maintenant le programme avec `while` et les fonctions `list` et `each`, utilisés à la place de `foreach` en PHP3.

```
while(list($clef, $valeur)=each($departements)){
   echo "$clef => $valeur<br>\n";
}
//Puis l'affichage uniquement des valeurs
while (list(, $valeur) = each ($departements)) {
   echo "$valeur<br>\n";
}
```

Comme les développeurs de PHP ont à cœur d'assurer une compatibilité ascendante, cette dernière forme est compatible avec PHP4 mais moins élégante.

Le traitement générique d'un formulaire

Dans cette partie qui explique la boucle `foreach` et `each… list`, profitons-en pour créer une application très simple afin de traiter un formulaire.

Le moteur PHP, quand les données d'un formulaire sont envoyées vers un fichier PHP, crée un tableau. Si vous utilisez la méthode `GET`, ce seraun tableau nommé `$HTTP_GET_VARS`, pour la méthode `POST`, ce sera `$HTTP_POST_VARS`. Pour envoyer les données, il suffit de faire une boucle sur le tableau :

```
while(list($clef, $valeur)=each($HTTP_POST_VARS)){
   $message.= "$clef : $valeur\n";
}
mail ("$votre_email","RESULTATS DU FORMULAIRE",
 "$message","from: monsite.com");
```

ou avec `foreach` en HTML :

```
foreach($HTTP_POST_VARS as $clef=>$valeur){
   $message.= "$clef : $valeur<br>\n";
}
mail ("$votre_email","RESULTATS DU FORMULAIRE",
 "$message","from: monsite.com");
```

Vous recevrez toutes les données du formulaire. Si certaines sont inutiles, il vous suffit d'ajouter dans la boucle :

Listing 3.13 : *Traitement pour tout formulaire : traitement_form_generic.php*

```
if ($clef="submit" || $clef="method"){
   continue;
}
```

Pour les tableaux `$HTTP_POST_VARS` et `$HTTP_GET_VARS`, voyez le chapitre *L'internaute est d'abord une personne*.

Break et continue

`break` permet de sortir de la boucle et `continue` perpétue la boucle en coupant l'exécution du bloc pendant le tour courant.

`break` permet de s'abstenir de limite supérieure dans la parenthèse de `for` car c'est `break` qui fixe cette limite à 2020 :

Listing 3.14 : *Les années bissextiles3 : bissextile3.php*

```
<?
for ($an=1582; ;$an++){
/* ici nous laissons la limite supérieure en blanc car
l'instruction break va nous faire sortir de la boucle
quand $an est égal à 2021 */
   if ($an==2021){
```

```
      break;
   }
   elseif ($an%100==0){
      if ($an%400==0){
         echo "$an<br>";
      }
   }
   elseif ($an%4==0){
      echo "$an<br>";
   }
}
?>
```

Reprenons l'exemple étudié dans la partie sur la boucle `for`. Au lieu de faire une condition pour inscrire l'année bissextile, nous allons définir une condition pour court-circuiter cette instruction avec la commande `continue` :

Listing 3.15 : *Les années bissextiles4 : bissextile4.php*

```
<?
for ($an=1582;$an<2020;$an++){
   if ($an%100==0){
      if ($an%400>0){
/////////si l'année n'est pas divisible par 400//
         continue;
      }
   }
   elseif ($an%4>0){
      ///ni par 4
      continue;
   }
   echo "$an<br>";
}
?>
```

3.5 Les fonctions de PHP

Vous avez ici le catalogue des fonctions les plus courantes de PHP. Pour les autres, vous pourrez consulter le manuel en ligne (voir annexe). Le but n'est pas de reproduire la documentation du langage PHP mais de vous proposer les fonctions les plus utiles.

Chaînes de caractères

Comme le type `string` est le type principal en PHP, vous serez souvent amené à utiliser les fonctions qui manipulent les chaînes de caractères. En voici quelques unes :

Tab. 3.8 : Les principales chaînes de caractères

Fonction	Exemple	Description
`addSlashes()`	`$a=addSlashes($a);`	Ajoute des caractères d'échappement.
`htmlentities()`	`$a=htmlentities ($a);`	Convertit les caractères locaux en caractères HTML, par exemple é en `é`.
`strip_tags()`	`$texte=strip_tags ($texte)`	Enlève les balises HTML et PHP.
`stripSlashes()`	`$a=stripSlashes ($b);`	Enlève les caractères d'échappement.
`strlen()`	`$nombre=strlen($a);`	Retourne la longueur de la chaîne.
`substr()`	`$b=substr($a,0,2)`	Retourne une sous-chaîne à partir d'une chaîne.
`trim()`	`$a=trim($a);`	Enlève les espaces blancs qui encadrent la chaîne.

htmlentities()

Nous vous conseillons d'utiliser cette fonction le moins possible car les navigateurs lisent très bien les caractères accentués, cela pour plusieurs raisons : le `strtoupper()` d'un caractère accentué crée une capitale accentuée. N'essayez pas par contre de mettre un caractère HTML en majuscule (exemple : `é` devient `&EACUTE;` et non É).

Si vous transformez le résultat d'un formulaire avec `htmlentities()`, vous serez obligé de faire l'opération contraire quand vous restituerez les données à l'intérieur d'un champ, dans le cas d'une modification des données, encore du code inutile.

Nous le verrons dans le chapitre intitulé *L'internaute est d'abord une personne*.

trim()

Vous utiliserez `trim()` pour une authentification afin que le mot de passe saisi soit débarrassé de tout caractère espace, avant ou après. Quand l'internaute entre des données, il utilise parfois la fonction couper/coller. Des espaces supplémentaires peuvent alors s'y insérer comme passagers clandestins.

addSlashes() et stripSlashes()

Utilisez, comme nous l'avons vu précédemment la fonction `phpinfo()` pour voir la configuration du serveur. Si `magic_quotes_gpc` est sur `On`, vous n'aurez pas besoin d'ajouter des caractères d'échappement à vos requêtes SQL. PHP les ajoutera automatiquement devant `"`, `'`, `\` et `NULL`. Par contre, si `magic_quotes_runtime` est sur `On`, vous devrez les enlever avec `stripSlashes()`, à chaque fois que vous extrairez des données de votre base *MySQL* ou autre.

Strlen

Vous utiliserez `strlen()` pour tester si vous affichez des données sur une page :

```
if ($fax){
   echo "Fax : $fax";
}
```

Ici, évidemment, vous n'affichez pas une ligne `Fax` si la personne n'a pas de fax.

substr()

Cette fonction vous permet d'extraire une chaîne de l'intérieur d'une chaîne. La première lettre est à l'indice `0`.

Vous pouvez par exemple épeler un nom :

```
$nom="Baudelaire";
$nombre_lettre=strlen($nom);
for ($i=0;$i<$nombre_lettre;$i++){
   $lettre=substr($nom,$i,1);
   echo "$lettre";
   if ($i==$nombre_lettre-1){
      echo ".";
   }
   else {
      echo ", ";
   }
}
```

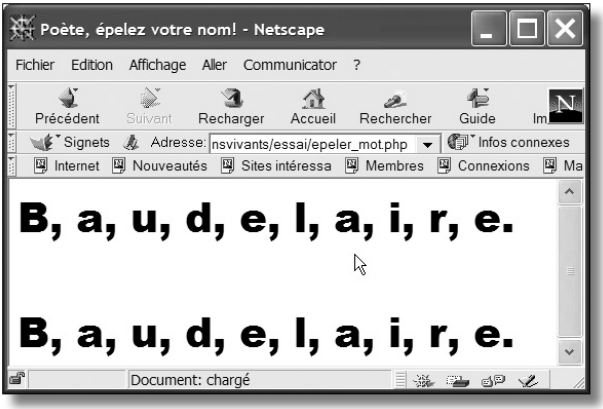
Vous pouvez également faire plus simple :

Listing 3.16 : *Épeler Baudelaire : epeler_mot.php*

```
$nom="Baudelaire";
$nombre_lettre=strlen($nom);
for ($i=0;$i<$nombre_lettre;$i++){
   $lettre=$nom[$i];
   echo "$lettre";
   if ($i==$nombre_lettre-1){
      echo ".";
   }
   else {
      echo ", ";
   }
}
```

Nous avons substitué `$lettre=$nom[$i];` à `$lettre =substr($nom,$i,1);` ce qui prouve que les chaînes de caractères sont des tableaux. Cette substitution n'aurait pas pu avoir lieu si nous avions extrait plus d'un caractère à chaque boucle (voir fig. 3.6).

Deux fonctions liées à la date et à l'heure sont très utiles. Pour ordonner des dates, il est pratique de les présenter sous la forme *AAAAMMJJ* ou *AAAAMMJJHHMM*. Ainsi, les dates les plus récentes composent un chiffre plus élevé. Nous allons élaborer deux fonctions de conversion de *AAAAMMJJ* à *JJ/MM/AAAA* et vice versa :



▲ Figure 3.6 : *Épelez votre nom*

```
<?
function nombre_date($cettedate){
   $ladate=$cettedate;
   $an=substr($ladate,0,4);
   $mois=(int)substr($ladate,4,2);
   $jour=(int)substr($ladate,6,2);
   $ladate=$jour.'/'.$mois.'/'.$an;
   if ($longueur>8){
      $heure= (int)substr($ladate,8,2);
      $minutes= substr($ladate,10,2);
      $ladate.=" ${heure}h ${minutes}mn";
      if ($longueur>12){
         $secondes= substr($ladate,12,2);
         $ladate.=" $secondes";
      }
   }
   return $ladate;
}
?>
```

Nous vérifions, dans cette fonction, la longueur de la date et, selon la taille, en extrayons aussi l'heure, les minutes et les secondes. Les jours, mois et heures sont forcés en `int` (typées) car nous voulons éviter qu'ils commencent par un 0 :

```
<?
function date_nombre($cettedate){
```

```
   $ladate=$cettedate;
   $ladate=explode('/',$ladate);
   $jour=$ladate[0];
   $mois=$ladate[1];
   $an=$ladate[2];
   $longueur=strlen($an);
   if ($longueur==2){
      if ($an>date("y")){
         $an='19'.$an;
      }
      else {
         $an='20'.$an;
      }
   }
   $longueur=strlen($mois);
   if ($longueur==1){
      $mois='0'.$mois;
   }
   $ladate=$an.$mois.$jour;
   return $ladate;
}
?>
```

Cette fonction crée d'abord un tableau à partir des éléments délimités par une barre oblique avec la fonction `explode()`. Puis nous vérifions la taille de chaque élément pour ajouter un 0 devant, en cas de besoin.

Autres fonctions

Voici d'autres fonctions de chaînes de caractères qui simplifient votre application.

Tab. 3.9 : D'autres fonctions de chaînes de caractères

Fonction	Exemple	Description
`chop()`	`$a=chop($a);`	Débarrasse la chaîne des espaces finaux.
`explode()`	`$tableau=explode(' ', $a);`	Crée un tableau de lettres séparées par un caractère.
`implode()`	`$a=implode(' ', $tableau);`	Retourne une chaîne à partir d'un tableau en y insérant un caractère de séparation.
`nl2br()`	`$a=nl2br($a);`	Adjoint aux caractères de nouvelle ligne des balises .

Tab. 3.9 : D'autres fonctions de chaînes de caractères

Fonction	Exemple	Description
`str_replace()`	`$a=str_replace($b, $c, $a)`	Remplace une sous-chaîne par une autre dans une chaîne.
`strip_tags()`	`$texte=strip_tags ($texte)`	Enlève les balises HTML et PHP.
`strtolower()`	`$a=strtolower($a);`	Met la chaîne en bas de casse
`strtoupper()`	`$a= strtoupper ($a);`	Met la chaîne en capitales.
`ucfirst()`	`$a=ucfirst($a);`	Met le premier caractère du mot en capitale.
`ucwords()`	`$a=ucwords($a);`	Met le premier caractère de chaque mot en capitale.

Variables

Les variables servent souvent de témoins dans vos applications et décident, selon leur existence ou leur valeur, de l'exécution d'un bloc de code.

Tab. 3.10 : Les variables

Fonction	Description
`empty()`	Évalue si la variable est non initialisée ou égale à 0.
`isset()`	Évalue si la variable est initialisée.
`unset()`	Désaffecte la variable.
`gettype()`	Retourne le type de la variable.
`is_array()`	Évalue si la variable est de type tableau.

Date et heure

Dans des applications interactives, vous appellerez souvent les fonctions de date et d'heure pour afficher l'heure et la date, soumettre une information à une date d'expiration, ordonner des listes en fonction de leur date, etc.

Tab. 3.11 : Les fonctions de date et heure

Fonction	Description
`string date (string format [, int timestamp])`	Formate un `timestamp` en année, mois, jour (`timestamp` : nombre de secondes depuis le 1/1/70).
`array getdate ([int timestamp])`	Décompose un `timestamp` en année, mois, jour … qu'il distribue dans un tableau.
`string microtime ()`	Retourne un millionième de seconde accompagné du `timestamp`.
`int mktime (int heure, int minute, int seconde, int mois, int jour, int année [, int is_dst])`	Fabrique un `timestamp` à partir d'une date.
`string strftime (string format [, int timestamp])`	Formate une date à l'aide des paramètres locaux.
`int time()`	Retourne le `timestamp` courant.
`string gmdate (string format [, int timestamp])`	Fonction `date()` pour le temps GMT
`int gmmktime (int heure, int minute, int seconde, int mois, int jour, int année [, int is_dst])`	Fonction `mktime` pour le temps GMT, `is_dst` est sur 1 si c'est une heure d'été.
`string gmstrftime (string format [, int timestamp])`	Fonction `strftime` pour le temps GMT

Formatage pour la fonction date()

La fonction `date()` utilise un code pour formater un `timestamp` :

Tab. 3.12 : Les codes de formatage de la fonction date()

Description	Option
Année en deux chiffres	`y`
Année en quatre chiffres	`Y`
Mois précédé d'un zéro comme "01" ou "02"	`m`
Mois sans adjonction de zéro comme "1", "12"	`n`

Tab. 3.12 : Les codes de formatage de la fonction date()

Description	Option
Jour de l'année comme "0" ou "335"	`z`
Jour du mois à deux chiffres (avec un 0)	`d`
Jour du mois sans adjonction de 0 comme "2", "30"	`j`
Jour numérique de la semaine (dimanche est 0 et samedi, 6)	`w`
Heure au format 12 précédée au besoin d'un zéro comme "01"	`h`
Heure au format 12 sans adjonction de zéro comme "2", "11"	`g`
Heure au format 24 précédée au besoin d'un zéro comme "01", "20"	`H`
Heure au format 24 sans adjonction de zéro comme "2", "15"	`G`
am ou pm (ante meridiem ou post meridiem : avant midi ou après midi)	`a`
Minutes de "00" à "59"	`i`
Secondes de "00" à "59"	`s`
Différence avec l'heure de Greenwich en heures comme "+0200"	`O`
Différence avec l'heure de Greenwich en secondes comme "+3600" ou "-7200"	`Z`
Numéro de la semaine dans l'année (la semaine commençant un lundi)	`W`
`timestamp` d'Unix (nombre de secondes depuis le 1/1/1970)	`U`
Nombre de jours dans le mois	`t`
Année bissextile si c'est égal à 1, sinon 0	`L`
Numéro de la semaine dans l'année (la première semaine commençant un lundi)	`W`
Nom du fuseau horaire sur lequel la machine est réglée, comme par exemple (CET)	`T`
`Daylight Savings Time` : 1 si c'est en période d'heure d'été, sinon 0	`I`
Nom du fuseau horaire sur lequel la machine est réglée, comme (CET)	`T`

À la fin de ce chapitre, une application vous permet d'afficher indifféremment votre heure, que ce soit en été ou en hiver, sans avoir de manipulation à faire. Cette application est pratique si votre serveur n'est pas dans votre pays car vous n'allez pas guetter l'heure pour la changer à 1 heure du matin un dimanche. Nous vous l'exposerons une fois que vous aurez appris à créer des fonctions. En voici le résultat :

▲ Figure 3.7 : *Date et heure*

`strftime()` formate une date en tenant compte des paramètres locaux de `setlocale()`.

Tab. 3.13 : Le code de formatage de la fonction strftime()

Description	Option
Année sur deux chiffres	%y
Année sur quatre chiffres	%Y
Date	%x
Date et heure	%c
Heure	%X
Heure notée am ou pm	%p

Tab. 3.13 : Le code de formatage de la fonction strftime()

Description	Option
Heure notée de 00 à 23	`%H`
Heure notée de 01 à 12	`%I`
Minute à deux chiffres	`%M`
Seconde à deux chiffres	`%S`
Jour de la semaine (le dimanche correspond à 0)	`%w`
Jour de la semaine (nom abrégé)	`%a`
Jour de la semaine (nom complet)	`%A`
Jour du mois de 00 à 31	`%d`
Jour de l'année 001 à 366	`%j`
Mois (nom abrégé)	`%b`
Mois (nom complet)	`%B`
Mois de 1 à 12	`%m`
Numéro de la semaine (l'année débute par un dimanche)	`%U`
Numéro de la semaine (l'année débute par un lundi)	`%W`
Fuseau horaire	`%Z`
Caractère `%`	`%%`

Mathématiques

La valeur PI

Tab. 3.14 : Valeurs mathématiques

Constante ou fonction	Valeur	Description
`M_PI`	3.1415926535898	Constante avec une valeur approximative en 14 décimales
`pi()`	3.1415926535898	Fonction identique

Les nombres pseudo-aléatoires

Ces nombres sont qualifiés de pseudo-aléatoires car ils sont générés par une simulation, à partir d'un algorithme mathématique. Ces fonctions génèrent des séquences temporelles de chiffres.

Tab. 3.15 : Nombres pseudo-aléatoires

Fonction	Description
`int rand ([int minimum], [int maximum])`	Entier aléatoire dans l'intervalle `[0..RAND_MAX]`
`void srand (int graine)`	Équivalent de `rand` mais générant les nombres avec une graine

Voici un petit programme pour générer un mot de passe aléatoire :

Listing 3.17 : *Création d'un mot de passe aléatoire : motdepasse.php*

```
<?
$alphabet = "abcdefghjkmnopqrstuvwxyz";
$alphabet .= "ABCDEFGHJKLMNOPQRSTUVWXYZ";
$alphabet .= "123456789";
/* vous remarquerez que certains éléments ont été omis pour
 ne pas créer d'ambiguïtés comme le chiffre 0 ou la
 minuscule l ou la majuscule I */

$nbcar = 11; $i = 0;$motdepasse = "";
// taille du mot de passe 11 caractères et initialisation

srand((double)microtime()*1000000);
/* initialisation du hasard avec le moment en microsecondes.
 Vous avez remarqué que le type a été forcé sinon nous aurions
 un entier qui serait égal à 0. La microseconde a une
 précision à 1/1 000 000. Ici vous plantez une graine qui est
 différente de 999999 autres graines possibles*/

while ($i<$nbcar) {
  $valcar = rand(0, strlen($alphabet));
  $motdepasse .= substr($alphabet,$valcar,1);
  $i++;
}
/*boucle pour générer les 11 caractères du mot de passe
par concaténation*/
?>
```

▲ Figure 3.8 : *Mot de passe*

3.6 Les fonctions

La fonction est une unité de programme enfermée à l'intérieur d'accolades avec comme entrée, des parenthèses. Nous verrons cependant qu'une variable déclarée `global` peut agir à l'intérieur de la fonction.

Le rôle de la fonction

En PHP, vous pouvez utiliser des fonctions c'est-à-dire de petits programmes dédiés à une tâche. Une fonction est autonome, elle reçoit ou non des paramètres et exécute son code, puis retourne le résultat à l'expression qui l'a appelée.

L'intérêt de ces programmes est que vous n'avez pas besoin de faire à chaque fois un couper/coller dans vos applications pour récupérer ces petits morceaux de programme, il vous suffit de les appeler. Ainsi, vos programmes sont plus lisibles. Quand vous traitez une variable dans une fonction, vous n'avez pas besoin de connaître le nom de la variable utilisée par la fonction.

Ici, nous créerons une fonction qui nous servira plus loin. Si vous voulez rendre votre site web interactif, invitez les visiteurs à s'inscrire dans votre club. Vous leur demandez leur nom et prénom et les remerciez ensuite. Vous souhaitez que votre site ait l'air professionnel, mais parfois, les personnes ne mettent pas leur nom en majuscules, ce qui fait désordre

quand vous les nommez dans les remerciements. Et puis le français n'est pas simple et vous ne voulez pas vous fâcher pour un nom écorché. La solution est d'y penser tout de suite pour ne plus y revenir. Vous fabriquerez donc une fonction traitant les noms de famille qui nécessitent des capitales.

Voici la fonction que nous vous engageons à mettre dans un fichier *fonctions.inc.php*. Nous verrons au cours du chapitre 4 comment l'utiliser dans un programme :

```
function majuscules($majuscule){
$majuscule=trim($majuscule);
$majuscule=strtolower($majuscule);
$majuscule=str_replace("d'", "d' ",$majuscule);
$passage=1;
///vous pouvez ajouter d'autres séparateurs dans le tableau///
$motif[]=" ";
$motif[]="-";
$nombre=count($motif);
```

Nous avons fourni les variables avec des valeurs. Nous allons signaler les différentes étapes avec le marqueur `$passage`. La boucle testera les séparateurs de mots pour vérifier qu'ils sont présents dans le paramètre fourni (`$majuscule`). La variable `$nom_complet` sert temporairement de transport aux éléments du nom décomposé. Une fois qu'elle a servi, elle rend son butin et s'efface avec la fonction `unset()` :

```
for ($i=0;$i<$nombre;$i++){
   $particule=$motif[$i];
   if (ereg($particule,$majuscule)) {
      if ($passage==2){
         $majuscule=$nom_complet;
         unset($nom_complet);
      }
      $passage=2;
      $majuscules=split($particule,$majuscule);
      $combien=count($majuscules);
```

Cette boucle `for` nichée dans la boucle principale gère les tronçons du nom coupé par le séparateur en mettant la première lettre en capitale.

```
      for ($j=0;$j<$combien;$j++){
         $maj=$majuscules[$j];
```

Nous plaçons la première lettre de chaque tronçon en majuscule, sauf pour ces mots-là :

```
        if ($maj!='de' && $maj!='d\'' && $maj!='et' &&
            $maj!='&'){
           $maj=ucfirst($maj);
        }
```

Nous reconstituons le nom complet au fil des boucles avec `$nom_complet` qui ne sert qu'au transport et qui, après une série de transformations, restitue sa valeur à la variable `$majuscule` :

```
        $nom_complet.=$maj;
        if ($j<$combien-1){
           $nom_complet.=$particule;
        }
```

Tant que nous n'en sommes pas au dernier tronçon, nous ajoutons le séparateur :

Listing 3.18 : *Mettre un nom en majuscules : majuscules.php*

```
    }
/////FIN DU PLUS PETIT FOR/////////

  }
////FIN DES MOTS QUI POSSÈDENT UN SÉPARATEUR " " ou -///////

 }
///////FIN DU PLUS GRAND FOR/////////
if($passage==1){
   $majuscule=ucfirst($majuscule);
}
elseif ($passage==2){
   $majuscule=$nom_complet;
}
//RESTITUTION DE LA VALEUR///

unset ($nom_complet);
///DESTRUCTION DU VÉHICULE AVEC UNSET////

$majuscule=str_replace("d' ", "d'",$majuscule);
////élision de l'espace supplémentaire pour d'///

return $majuscule;
}
```

La fonction commence par le mot-clé `function`. Ensuite, `trim()` enlève les espaces éventuels qui encadreraient le nom puis met la chaîne en bas de casse. Le `str_replace()` est présent pour introduire une astuce car un nom à particule peut être composé d'un `de` ou `d'` qui ne prend pas de capitales. Ici, la fonction introduit un espace après `d'` pour l'assimiler à `de`. Il suffira ensuite de retirer l'espace pour que le nom soit correct.

Les lignes qui suivent insèrent les séparateurs (ici espace ou tiret) dans un tableau. Il est possible d'ajouter des séparateurs car la fonction `count()` qui suit donne le nombre d'éléments, donc de boucles à faire pour traiter le nom.

Nous entrons dans deux boucles, la première va traiter le nom en regard de chaque séparateur inséré dans le tableau. La deuxième est imbriquée dans la première, elle intervient pour traiter les tronçons de noms avant de les reconstituer. À chaque tour de la grande boucle, il est nécessaire de vider la variable de sa valeur de résultat, pour éviter plusieurs concaténations du même nom d'un séparateur à l'autre. Pour cela le marqueur `$passage` se révèle nécessaire.

▲ Figure 3.9 : *Mettre en majuscules*

Reportez-vous au chapitre *Moteurs de recherche et expressions rationnelles* pour les expressions rationnelles incluses dans cette fonction.

La portée d'une variable dans une fonction

Une variable à l'extérieur d'une fonction possède une portée limitée à la périphérie de la fonction, c'est-à-dire que la variable n'existe pas dans ce périmètre. Seuls un ou plusieurs paramètres seront envoyés dans la fonction par l'intermédiaire des parenthèses, mais le nom de la variable n'a aucune importance. C'est d'ailleurs ce qui fait son intérêt.

Parfois cependant, il est intéressant de récupérer une variable qui existe dans la même page en dehors de la fonction. Pour cela, il est possible d'utiliser le mot-clé `global` afin que la portée de la variable se situant à l'extérieur de la fonction soit étendue à toute la page.

Dans l'exemple cité précédemment, il est possible de mettre les séparateurs avant la fonction pour une meilleure visibilité :

```
$motif []=" ";
$motif []="-";
```

Ensuite, dans la fonction vous déclarez :

```
global $motif;
```

Ainsi, la fonction pourra utiliser le tableau `$motif` même s'il est situé à l'extérieur de la fonction. Précédé du mot `global`, il est opérationnel dans la fonction.

Méfiez-vous des variables globales

Si vous modifiez votre variable et que vous avez oublié qu'elle est opérante dans la fonction, vous risquez d'avoir une surprise dont il sera difficile de déceler la cause.

Les fonctions imbriquées

Il est intéressant, parfois, de créer une fonction pour appeler plusieurs fonctions. Ainsi, lorsque vous relirez votre programme, vous obtiendrez une sorte de hiérarchie de fonctions.

Nous pourrions avoir, par exemple :

```
function verif_id ($login, $motdepasse, $table){
   connexion_base();
   recherche_dans_table($table, $login, $motdepasse);
}
```

Les fonctions récursives

Une fonction peut s'appeler elle-même, de façon récursive. Nous allons le voir avec une suite de Fibonacci. Pour la petite histoire, Leonardo Fibonacci était un moine italien qui vivait 250 ans avant Léonard de Vinci. Il avait remarqué que les végétaux respectaient une suite dans leur croissance. Ainsi, les pétales de fleurs ou les feuilles sur les arbres observaient une sorte de règle. Il en fit une fonction mathématique. Cette fonction génère un nombre en faisant la somme des deux précédents. Les deux premiers sont 0 et 1 puis 0+1=1; 1+1=2; 2+1=3 ; 3+2=5… Si vous continuez, vous parviendrez rapidement à des nombres vertigineux.

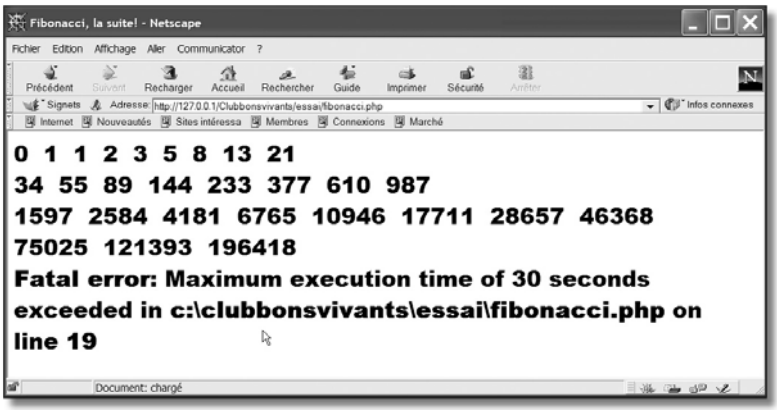
Voici la fonction :

Listing 3.19 : *Fonction suite de Fibonacci fibonacci.php*

```
<?
function fibonacci ($nombre){
   if ($nombre<0){
      return (0);
   }
   elseif($nombre<2){
      return ($nombre);
   }
   return fibonacci($nombre-1)+fibonacci($nombre-2);
}

for ($i=0;$i<20;$i++){
   $nouveau_nombre=fibonacci($i);
   echo $nouveau_nombre;
   echo '<br>';
}
?>
```

Les deux premiers tests avec `if` et `elseif` concernent les deux premiers nombres qu'il faut générer, puis la fonction s'appelle deux fois et retourne

le résultat. La boucle `for` est là pour créer la liste des nombres obtenus. Le calcul au-delà de 30 dépend de la capacité de votre machine.



▲ Figure 3.10 : *La fonction de Fibonacci*

Afficher vos date et heure locales

Le programme proposé ici est l'occasion de revoir des fonctions imbriquées ainsi que d'autres notions telles que les dates, les tableaux et l'opérateur de contrôle d'erreur.

Pour afficher le jour de la semaine en français, nous créons un tableau car le jour est un nombre de 0 à 6 en partant du dimanche. Le paramètre envoyé `$moment` est un `timestamp` que nous calculons avant l'appel. Nous tenons compte du décalage avec l'heure GMT pendant l'heure d'hiver. L'heure d'hiver commence le dernier dimanche d'octobre à 1h GMT et finit le dernier dimanche de mars à 1h GMT.

```
<?
function affiche_jour($moment){
   $jour=date("w", $moment);
   $les_jours=array('dimanche', 'lundi', 'mardi', 'mercredi',
                    'jeudi', 'vendredi', 'samedi');
   $jour=$les_jours[$jour];
   return $jour;
}
```

Cette fonction affiche_mois() affiche le mois en français (pas de capitales en français).

```
function affiche_mois ($moment){
   $mois=date("n", $moment);//le mois sans 0 devant
   $les_mois=array('', 'janvier', 'février', 'mars',
                   'avril', 'mai', 'juin', 'juillet',
                   'août', 'septembre', 'octobre',
                   'novembre', 'décembre');
   $mois=$les_mois[$mois];
   return $mois;
}
```

Voici maintenant la fonction principale qui contient l'algorithme et appelle les autres fonctions pour se nourrir :

```
function affiche_date_locale($moment){
   $decalage_hiver=1;
```

Nous vérifions d'abord si un paramètre a été envoyé. Si ce n'est pas le cas, nous prenons le TIMESTAMP Unix courant :

```
if (!$moment){
   $temps=time();
}
else {
   $temps=$moment;
}
```

Nous créons les TIMESTAMP Unix des moments du changement d'heure pour l'année en cours en heure GMT. Il suffit de trouver le jour de semaine du 31 mars, puis en sachant que le dimanche est un jour 0, d'enlever le nombre du jour de semaine à 31 pour obtenir la date :

```
$jourdesemaine=gmdate("w", mktime(1,0,0, 3,31,gmdate("Y")));
//variable pour connaître le jour de semaine du 31 mars

$limite_inf=mktime(1,0,0, 3,31-$jourdesemaine,gmdate("Y"));
/*variable pour trouver le dernier dimanche de mars pour l'année
courante*/

$jourdesemaine=gmdate("w", mktime(1,0,0, 10,31,gmdate("Y")));
/*variable pour connaître le jour de semaine du 31 octobre*/

$limite_sup=mktime(1,0,0, 10,31-$jourdesemaine,gmdate("Y"));
```

```
/*variable pour trouver le dernier dimanche d'octobre pour l'année
courante*/
```

Maintenant, nous testons la date pour savoir si elle est comprise entre les deux limites. Dans ce cas, elle est en heure d'été :

```
if ($temps>$limite_inf && $temps<$limite_sup){
   $decalage=$decalage_hiver+1;
}
else {
   $decalage=$decalage_hiver;
}
$moment=mktime(gmdate("G")+$decalage,gmdate("i"),0,
               gmdate("n"),gmdate("j"),gmdate("Y"));
```

Enfin, nous fabriquons le `timestamp` avec `mktime()`, en incluant le décalage avec l'heure GMT :

Listing 3.20 : *Fonctions d'affichage de date : date.php*

```
$ladate=affiche_jour($moment) ." ".date("j", $moment)." ".
        affiche_mois($moment) ." ".  date("Y", $moment);
/*la date inclut le mois en français grâce à l'appel de
la fonction.*/

$heure = date("G",$moment);
$minute = date("i", $moment);
$ladate.="<BR>";
$ladate.="$heure";
$ladate.="h";
$ladate.="$minute";
return $ladate;
}
echo @affiche_date_locale();
/*Vous faites précéder la fonction du @ pour éviter le
 message d'erreur si vous n'envoyez pas de paramètre*/
?>
```

3.7 Récapitulatif pour votre application

Vous avez créé un fichier *fonctions.inc.php* dans lequel vous avez inséré votre fonction `majuscules()`. Maintenant, vous allez si vous le désirez y insérer les fonctions `date_nombre()`, `nombre_date()`, `affiche_date_locale()`, `affiche_mois()` et `affiche_jour()`. Nous vous encourageons à les modifier comme bon vous semble, pour les adapter dans leur

forme d'affichage ou leur contenu. Si vous vivez à Montréal et que votre serveur est sur un autre fuseau horaire, votre décalage avec l'heure GMT sera plutôt -5 en hiver que +1, etc.

Nous verrons dans le chapitre 4 comment insérer ces fonctions dans les fichiers de votre application avec une seule ligne de code. Après la clôture de ce chapitre, nous étudierons le traitement des formulaires et leur vérification.

4

MySQL en action

Nous allons bientôt pouvoir nous aventurer en ligne. Mais avant cela, revenons à notre internaute qui souhaite créer son club des Bon Vivants. Il va lui falloir mettre de l'ordre dans ses idées car avec *MySQL*, il passe à la vitesse supérieure.

MySQL a été créée par Michael Widenius de la société TcX, en réaction à certaines insuffisances de la base de données *mSQL*. La version 1.0 sort en mai 1995. Son créateur décide de la distribuer gratuitement via Internet.

L'emploi d'une base de données est lié à l'utilisation des formulaires et à la nécessité de classer les informations données par l'internaute. Il est bien sûr possible d'employer un simple fichier texte avec un séparateur entre les champs mais cette solution comporte plusieurs inconvénients.

4.1 Le fichier face à la base de données

Le fichier est fragile. Vous devez créer un double de votre fichier car tout fichier peut être vidé par un accès simultané. Il faut donc en verrouiller l'accès avec un verrou logiciel que le langage PHP fournit.

Votre séparateur doit être un signe que vous n'utilisez jamais, pour qu'on ne le confonde pas avec un élément de votre texte.

La base de données est gratuite et proposée par tous les hébergeurs. Elle est robuste (en particulier *MySQL*) et rapide. De plus, elle propose des fonctions qui économisent des lignes de code. L'application de ce chapitre vous permettra de mieux comprendre tout cela.

4.2 Conception de l'application du club

Vous disposez d'une base de données *MySQL* vide. Une base de données sert à classer les informations en vue de les retrouver facilement et d'opérer des rapprochements entre elles afin d'avoir des statistiques.

Les données sont ce qu'il y a de plus important. Tout développeur commence par là. Le site n'est ensuite là que pour présenter et traiter ces données.

La première information concerne l'identité des membres et leurs coordonnées :

- nom,
- prénom,

- adresse,
- code postal,
- ville,
- pays,
- e-mail,
- téléphone.

Tout le monde n'aura pas envie de laisser une adresse postale ou un téléphone, nous pouvons donc penser que ces informations sont à considérer comme optionnelles. Par contre, il est intéressant de disposer de la ville et du code postal car le but de ce club est la convivialité. Il se peut en effet que des membres soient proches géographiquement et puissent ainsi organiser des rencontres locales. Il faut donc rendre obligatoire le code postal et le pays.

L'adresse e-mail est obligatoire pour les contacts entre le webmestre et les membres du club, mais aussi pour les contacts entre les membres du club. Le prénom est intéressant et le nom optionnel.

Voyez également si l'adresse postale et le téléphone sont indispensables. Les contacts peuvent se faire par courriel. S'ils le désirent, les membres s'échangeront ces deux informations.

Le problème est que le membre du club qui change d'adresse e-mail (ce qui peut arriver souvent) demandera éventuellement au webmestre d'effectuer le changement. Cela peut prendre du temps si le club comprend de nombreux membres. Il est plus intéressant de permettre au membre un accès direct à ses données pour qu'il puisse, au besoin, les modifier. Mais comment être sûr qu'il s'agit de la personne en question et non d'une autre ? La solution ici passe par l'authentification. Nous l'étudierons au chapitre 5. Il faut ajouter un nom d'usage et un mot de passe. Le nom d'usage apporte un anonymat et le mot de passe garantit l'identité de la personne.

Créons maintenant ensemble la table dans la base de données.

4.3 Création d'une table

Sachez qu'il est intéressant dans une table de créer une clé unique qui identifie chaque nouvelle ligne. *MySQL* propose une fonction

AUTO_INCREMENT qui permet de créer automatiquement une clé qui s'incrémente de 1 à chaque nouvelle ligne. Chaque nom d'usage dans le club sera unique puisqu'il identifiera un membre, donc nom_usage peut servir de deuxième clé, de clé secondaire.

Chaque clé auto-incrémentée est unique

Chaque nouvelle ligne dispose d'une clé unique. Si vous effacez une ligne par la suite, la clé disparaîtra aussi.
Par exemple, vous avez une ligne avec une clé 1 puis vous l'effacez. Vous n'avez donc plus aucune ligne dans votre table mais la prochaine ligne commencera néanmoins à 2. Ainsi, l'intégrité des données est garantie.

Pour créer la table, nous devons définir le type de chaque donnée et sa longueur.

Le nom d'usage est une chaîne de caractères dont nous pouvons limiter la longueur à 25 caractères. Le mot de passe et le prénom sont de même type, avec une longueur de 20 pour le mot de passe et de 25 pour le prénom.

La longueur d'un champ MySQL

C'est à vous de décider de la longueur du champ en question. Pour un nom d'usage, vous pouvez imaginer qu'ils ne dépasseront pas 25 caractères, ce qui est assez long. C'est vrai qu'il vaut mieux calculer plus large car si la donnée est trop longue, elle sera tronquée dans la table. D'autre part, les hébergeurs vous accordent beaucoup de place et il ne sert à rien d'économiser dans ce domaine, pour être obligé ensuite d'élargir et de reconstituer les données tronquées.

Le code postal est constitué en France d'une série de cinq chiffres. Cette donnée peut donc être de type numérique. Il vaut mieux lui donner un type "chaîne de caractères" car des Québécois seront peut-être intéressés par votre club (leur code postal est composé de lettres et de chiffres). D'ailleurs, les neuf premiers départements français sont précédés d'un 0. En règle générale, les codes postaux, les numéros de téléphone… sont de type texte car ils ne sont pas des nombres mais des codes. Nous pourrions

utiliser les accents pour les noms de champ *MySQL*. Nous les éviterons cependant pour préserver l'homogénéité des noms.

L'homogénéité des noms

Il est plus simple de gérer des noms homogènes. Ainsi, vous pouvez utiliser des _ ou des capitales pour nommer un champ mot de passe : *mot_de_passe*, *MotdePasse*, *motdepasse* ou *mot_passe*. Choisissez un système de nommage et conservez-le. Des erreurs peuvent être générées par les noms. Si le champ d'une clé auto-incrémentée s'appelle toujours clef, vous n'aurez pas besoin de vérifier le nom à chaque programme.

Récapitulons la liste des données :

- `clef` numérique `AUTO_INCREMENT` 11 (caractères) ;
- `nom_usage` texte 25 ;
- `mot_passe` texte 20 ;
- `prenom` texte 30 ;
- `code_postal` texte 12 ;
- `ville` texte 35 ;
- `pays` texte 35 ;
- `email` texte 50.

Nous avons tous les éléments pour créer notre table avec la commande **CREATE** de *MySQL* :

```
CREATE TABLE membres (
clef int(11) NOT NULL auto_increment,
nom_usage varchar(25) NOT NULL default '',
mot_passe varchar(20) NOT NULL default '',
prenom varchar(30) NOT NULL default '',
code_postal varchar(12) NOT NULL default '',
ville varchar(35) NOT NULL default '',
pays varchar(35) NOT NULL default '',
email varchar(50) NOT NULL default '',
PRIMARY KEY (clef),
INDEX (clef),
UNIQUE KEY clef (clef),
)
```

Analysons maintenant cette commande. Nous avons le nom de la colonne ou champ, le type de données, la longueur, l'attribut `NULL` ou `NOT NULL`, l'attribut `DEFAULT`. Nous verrons la clé primaire, l'index et l'attribut `UNIQUE` avant de revenir plus précisément à notre club.

`NULL` est une absence de valeur, ce qui est différent d'une valeur ou de 0. L'option `NOT NULL` oblige à indiquer une valeur.

`DEFAULT` spécifie une valeur pour un champ même s'il a été oublié au cours d'un `INSERT` dans le cas d'un champ comportant la spécification `NOT NULL`.

La clé primaire évite les doublons dans la table. L'inconvénient est que vous ne pouvez la changer une fois qu'elle est choisie, d'où le choix d'une clé auto-incrémentée. Ainsi à chaque insertion de données, vous n'aurez pas besoin de créer une clé, elle prendra automatiquement la valeur de la ligne précédente augmentée de 1 ou, par défaut, la valeur de 0 plus 1. Cette clé indexée et unique permet une recherche rapide à l'intérieur de la table. L'attribut `UNIQUE` désigne des clés secondaires. Ainsi les champs *nom_usage* et *mot_passe* ne peuvent comporter de doublons. Chaque membre possède bien sûr une identité unique. Cependant, nous n'avons pas jugé intéressant de les caractériser ainsi dans la table *MySQL*. Il vaut mieux continuer dans l'artisanat et faire la vérification "à la main", par quelques lignes de code.

Vérification des doublons sur `nom_usage` et `mot_passe`

Dans notre exemple, le formulaire comprend deux pages, donc les données sont insérées dans la table en deux fois. Entre les deux pages, les champs *nom_usage* et *mot_passe* sont vides. Imaginez que deux membres s'inscrivent en même temps. Le deuxième membre ne pourra pas insérer ses données car le champ sera vide dans les colonnes `nom_usage` et `mot_passe` pour les deux lignes, donc considéré par *MySQL* comme un doublon.

L'indexation améliore les temps de réponse d'une table. La clé primaire est systématiquement indexée pour permettre à *MySQL* de vérifier si elle existe déjà dans la table lors d'une insertion. Chaque champ avec la caractéristique `UNIQUE` comprend un index.

4.4 Modification d'une table

En imaginant votre site, vous vous apercevez que vous aimeriez ajouter la date d'inscription de chaque membre.

Avec *MySQL*, deux options existent pour la date. Soit vous entrez une date, soit vous utilisez un format qui la crée automatiquement. Cette dernière option évite de créer une donnée pour l'insérer et économise du code. Il s'agit du type `TIMESTAMP`. L'inconvénient majeur de ce format de colonne est que si vous modifiez une donnée dans la ligne, comme par exemple l'e-mail, le champ modifiera aussi la date. C'est pourquoi nous conseillons l'autre option de date.

Nous pouvons mettre les deux types de colonne. Ainsi, nous saurons si le membre du club a modifié ses données, donc s'il est actif dans le club.

Pour insérer ces deux champs, nous utiliserons la commande **ALTER**. Les champs s'appelleront *date_inscription* et *date_misajour* :

```
ALTER TABLE membres
ADD date_inscription DATE not null,
ADD date_misajour TIMESTAMP(8) not null
```

Vous employez les commandes **ALTER TABLE** et **ADD** pour ajouter des colonnes à la table. Le type `DATE` a un format *AAAA-MM-JJ* par défaut et `TIMESTAMP` possède une longueur de 14 caractères par défaut, sous la forme *AAAAMMJJHHMMSS*. Ici, la longueur de 8 induit un format de type *AAAAMMJJ*.

Cette fois, vous commencez à voir votre site. Vous imaginez les membres qui s'inscrivent du monde entier à votre club. Vous rencontrez des membres d'autres pays et voyagez. Vous avez un cousin qui vit au Québec et vous voyez déjà parti vers ce pays pour rencontrer des Québécois inscrits sur le site. Vous pourrez également vous amuser à faire des statistiques sur la nationalité des membres. Attention, certains membres vont orthographier différemment le nom de leur pays, vos statistiques seront donc faussées. Grande-Bretagne pourrait être écrit Great Britain, Grande Bretagne, Royaume-Uni, United Kingdom… Certains Québécois

écriront Québec et d'autres Canada. Si le nombre de membres est important, la base deviendra difficile à gérer. Les clés étrangères sont une solution.

4.5 Les clés étrangères

La meilleure façon de créer une homogénéité est de proposer des choix à travers des menus déroulants. Ainsi l'internaute ne tape rien. Il choisit et n'entraîne pas d'erreur.

Nous allons créer dans la base une deuxième table renfermant tous les pays du monde. Cette table est composée de deux colonnes. La première est un code de trois lettres, la seconde le nom du pays.

Tables pour menus déroulants

Pour homogénéiser les tables qui contiennent les éléments des menus déroulants à l'intérieur d'un formulaire, elles auront toutes deux colonnes communes : code et nom.

```
CREATE TABLE pays (
  code char(3) NOT NULL default '',
  nom varchar(100) NOT NULL default '',
PRIMARY KEY  (code),
UNIQUE KEY code (code)
)
```

Vous trouverez les données pour créer une table sur le site `www.microapp.com`. Les trois lettres garantissent un code unique. Ainsi, dans la table *membres*, le champ *pays* renfermera une donnée texte d'une longueur fixe de 3. Vous écrivez donc dans le champ *requête* de PHPmyAdmin, la commande :

```
ALTER TABLE membres
CHANGE pays pays CHAR (3) not null
```

Précédemment, nous avions employé `ADD`. Nous changeons maintenant la nature de la colonne avec `CHANGE`.

Ainsi la donnée insérée dans la table *membres* et concernant le pays est une clé étrangère car elle a comme référence, la clé primaire d'une autre table. La clé étrangère est prise en compte avec la version 4.1 de *MySQL* (Nous en sommes à la 3.23).

Pour afficher le contenu de la table dans une page avec le code de la table *membres* remplacé par le nom de la table *pays*, nous utiliserons une jointure.

4.6 Les jointures de tables

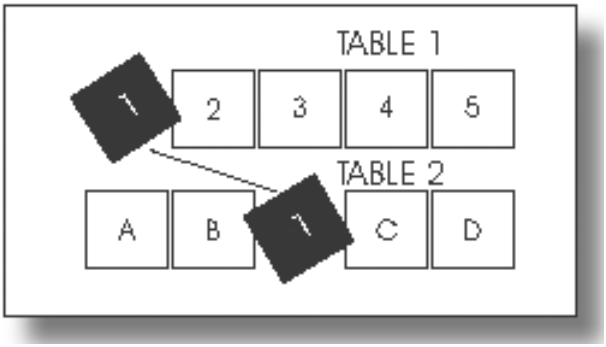
Pour afficher un résultat lisible, vous utilisez la requête `SELECT` en y incluant une jointure de table :

```
SELECT * from membres, pays where membres.pays=pays.code
```

Ici le signe * indique que tous les champs sont sélectionnés. Pour prendre en compte uniquement certains champs, il faut les nommer. Si les tables ont deux champs qui portent le même nom, il est nécessaire de nommer aussi les tables. Nous conseillons de le faire systématiquement, dans le cas des jointures, pour ne pas rencontrer de mauvaises surprises. Ainsi pour sélectionner tous les champs sauf la date de mise à jour, il faudrait écrire la requête suivante :

```
SELECT membres.nom_usage,
membres.mot_passe,
membres.prenom,
membres.code_postal,
membres.ville,
pays.nom ,
membres.email
from membres, pays where membres.pays=pays.code
```

Le point marque l'appartenance de `nom_usage` à la table *membres* par exemple.



▲ Figure 4.1 : *Jointure*

4.7 Les types de données MySQL

Récapitulons les types de contenu dans une table *MySQL*. Nous utilisons surtout les trois premiers types car dans l'application PHPmyAdmin, le champ de type *text* peut être lu alors que le champ de type *blob* ne pourra afficher son contenu. Les champs de type *char* occupent une place identique et contiguë tandis que les champs de type *varchar* sont dispersés dans l'amas des données et demandent plus de ressources serveur.

Tab. 4.1 : Les types texte et binaire

Type	Taille en bytes	Description
`char(n)`	Longueur	Un champ de texte d'une longueur fixe. Les chaînes plus courtes sont complétées (1 à 255).
`varchar(n)`	Longueur	Un champ de texte d'une longueur maximum de n (1 à 255)
`text`	Longueur + 2	Un champ de texte d'une longueur maximum de 64Kb
`blob`	Longueur + 2	Un champ binaire d'une longueur maximum de 64Kb

Pour le commerce électronique, les prix seront stockés dans des colonnes de type `DECIMAL`. Les différences de taille permettent d'économiser de la

place, ce qui est intéressant uniquement si vous avez de grandes quantités de données car alors, un octet de différence sur plusieurs centaines de milliers de champs peut créer une légère différence de temps de réponse et un gain de place sur le disque dur du serveur.

Tab. 4.2 : Les types numériques

Type	Taille	Description
`TINYINT(n)`	1	Entier non signé (`unsigned`) de 0 à 255, signé de -128 à 127
`SMALLINT(n)`	2	Entier non signé de 0 à 65535, signé de -32768 à 32767
`MEDIUMINT(n)`	3	Entier non signé de 0 à 16777215, signé de -8388608 à 8388607
`INT(n)`	4	Entier non signé de 0 à 4294967295, signé de -2147483648 à 2147483647
`BIGINT(n)`	8	Entier non signé de 0 à 4294967295, signé de -2147483648 à 2147483647
`FLOAT(n, dec)`	4	Nombre à virgule flottante à simple précision correspondant au type de même nom dans le langage PHP
`DOUBLE(n, dec)`	8	Nombre à virgule flottante à double précision correspondant au type de même nom dans le langage PHP
`DECIMAL(n, dec)`	Longueur	Nombres précis stockés sous forme de caractères

Méfions-nous de `TIMESTAMP`. Il vaut mieux l'utiliser seulement quand la date de mise à jour est la plus intéressante.

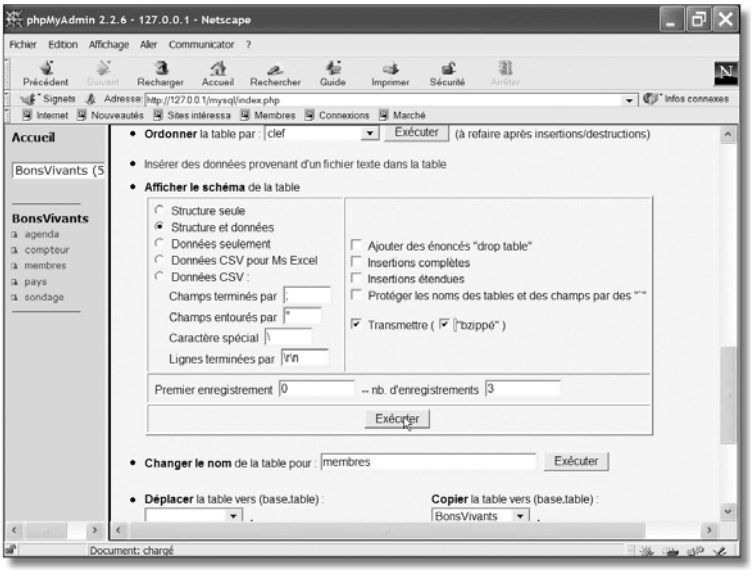
Tab. 4.3 : Les types date et heure

Type	Taille	Description
`TIMESTAMP(n)`	4	Mis à jour à chaque modification de la ligne. Si `n=14`, le nombre produit sera de type *AAAAMMJJHHMMSS*, pour 8, *AAAAMMJJ*, pour 10, *AAAAMMJJHH*…
`DATE`	3	Formaté *AAAA-MM-JJ* même si la date est entrée sous la forme *AAMMJJ*…

Tab. 4.3 : Les types date et heure		
Type	Taille	Description
TIME	3	Formaté *HH:MM:SS* même si l'entrée est sous la forme *HHMMSS* ou *HHMM*…
DATETIME	8	Formaté *AAAA-MM-JJ HH:MM:SS* quel que soit son format d'entrée

4.8 La sauvegarde de la base

Transportons maintenant la base de données de votre machine sur votre nouveau site.



▲ Figure 4.2 : *Sauvegarde de la base de données*

Avec PHPmyAdmin, il est simple et pratique de sauvegarder la base. Il vous suffit d'aller dans la rubrique *Afficher le schéma de la base*. Vous avez alors le choix de son contenu, soit uniquement la structure, la structure et les données, ou encore les données seulement. Ensuite, vous

avez la possibilité de d'enregistrer la base sur votre disque dur en fichier texte ou compressé. Faites une sauvegarde régulière. Sinon, en cas de problèmes, vous ne récupérerez peut-être pas les internautes inscrits depuis la dernière sauvegarde.

4.9 Les formulaires

Une des principales utilisations de PHP et raisons de son succès est la facilité qu'il procure au webmestre pour traiter les données envoyées par un formulaire. Le formulaire, comme le courriel, est un des vecteurs principaux de la communication entre les internautes et le webmestre du site.

PHP gère les données du formulaire de manière transparente. Il crée automatiquement les variables à partir des noms utilisés pour qualifier les champs du formulaire. Il entrepose ces variables dans un tableau de portée globale. C'est-à-dire que ces variables peuvent être appelées directement depuis n'importe quelle partie de l'application.

L'inscription au club

Dans une page d'accueil, vous ferez la promotion de votre club. Elle contiendra un lien qui permettra de rejoindre la page d'inscription. Cette dernière sera un formulaire.

Les visiteurs d'un site sont méfiants, à juste titre. Ils peuvent hésiter à donner leurs coordonnées à quelqu'un qu'ils ne connaissent pas. Rassurez-les sur la confidentialité de ces données et la possibilité pour eux de supprimer leur profil, le cas échéant.

Le fichier de connexion à la base

Nous avons vu précédemment que pour accéder à la base de données, il fallait un nom d'utilisateur et un mot de passe. Si vous avez besoin de cet accès dans votre page, saisissez les lignes suivantes :

```
<?
$dbname   = 'nom_base';
$hostname = 'localhost';
$username = 'nom_utilisateur';
$password = 'mot_de_passe';

if (!$id_link = mysql_connect($hostname, $username, $password)) {
    echo 'Connexion impossible à mysql';
```

```
    exit;
}

if (!mysql_select_db($dbname, $id_link )) {
    echo 'Sélection de base de données impossible';
    exit;
}
?>
```

Ces quelques lignes composent votre fichier de connexion. Sur votre site, vous allez créer un répertoire nommé *commun*. Dans ce répertoire, vous placerez tous vos programmes (fichiers inclus et fonctions). Ce fichier qui est un fichier inclus dans toutes les pages demandant une connexion à la base de données, s'appellera *connexion.inc.php*. `inc` est l'abréviation de `include`. Ce type d'architecture est pratique car elle facilite la migration d'une application d'un serveur vers un autre. Ainsi, il vous sera facile de transporter votre application, de votre serveur de test sur votre machine personnelle à celui de l'hébergeur, sans faire de changement majeur. Il vous suffira de modifier ce fichier.

Les fichiers inclus

Ils sont appelés depuis un autre fichier PHP avec la ligne `include 'commun/connexion.inc.php'`. Il est nécessaire dans ce type de fichier d'utiliser les balises PHP, bien qu'il soit inclus dans un fichier qui en comporte déjà.

La fonction `mysql_connect()` renvoie l'identifiant de connexion *MySQL* en cas de succès et `FALSE` si la connexion n'a pu se faire. Par les deux conditions (`if`…), l'action est générée et si la fonction renvoie `FALSE`, le bloc conditionnel est exécuté. Le message affiché en cas d'échec de la connexion ou de la sélection de la base de données est minimal, il vous appartient de le personnaliser et de l'agrémenter.

Exit

N'oubliez pas la commande **exit** à la fin de votre bloc conditionnel, sinon le reste de l'application sera exécuté et vous afficherez deux pages superposées.

La première page

Créons maintenant la page contenant un formulaire d'inscription au club. Nous ne ferons aucune innovation, nous reprendrons simplement le nom des colonnes de la table pour les champs du formulaire, sauf celles qui génèrent des données automatiques comme `clef` avec la fonction `AUTO_INCREMENT` et `date_misajour` avec le format *TIMESTAMP*, ainsi que celles qu'il nous appartient de renseigner comme `date_inscription`. Voici les colonnes :

`nom_usage`, `mot_passe`, `prenom`, `code_postal`, `ville`, `pays` et `email`.

La question est de savoir maintenant s'il faut déployer ce questionnaire sur plusieurs ou sur une seule page. Ne perdez pas de vue qu'il vaut mieux éviter d'effrayer l'internaute en lui présentant trop de champs à renseigner.

Faut-il générer un mot de passe ou laisser l'internaute le choisir ? Un mot de passe généré aléatoirement sera difficilement mémorisé par l'internaute mais mieux sécurisé. Le but est de mettre à l'aise le visiteur et de n'opposer aucun obstacle à son désir de participer. Le choix sera donc arrêté pour un mot de passe choisi par l'internaute. Cependant, il devra comporter au moins huit caractères.

La deuxième page du formulaire sera consacrée à la création du nom d'usage et du mot de passe.

Les pages de votre application

L'objet de ce livre n'est pas de créer de belles pages au design séduisant mais de décrire et expliquer des applications en PHP, pour que vous soyez capable de les créer. Nous vous conseillons d'améliorer vos pages avec un beau graphisme, une ergonomie de navigation claire et un ton personnel.

Voici la première étape d'écriture du code de la première page :

```
<html>
<head>
  <title>Inscription</title>
</head>
<body>
```

```
Pour vous inscrire dans le club, nous avons besoin de quelques
renseignements. Préparez un nom d'usage pour le club et un mot de
passe de plus de 8 caractères dont vous vous souviendrez facilement.
<form action="inscription_p2" method="post" name="page_1>
<table cellspacing="2" cellpadding="2" border="0">
<tr>
 <td>Votre prénom :</td>
 <td><input type="text" name="prenom"></td>
</tr>
<tr>
 <td>Votre adresse email</td>
 <td><input type="text" name="email"></td>
</tr>
<tr>
 <td>Votre code postal</td>
 <td><input type="text" name="code_postal"></td>
</tr>
<tr>
 <td>La ville où vous vivez</td>
 <td><input type="text" name="ville"></td>
</tr>
<tr>
 <td>Votre pays</td>
 <td><input type="text" name="pays"></td>
</tr>
<tr>
 <td></td>
 <td><input type="submit" name="Je m'inscris"></td>
</tr>
</table>
</form>
</body>
</html>
```

Nom des champs et nom des colonnes

Vous constatez que l'attribut `name` des champs du formulaire reprend la dénomination des colonnes de la table, voici pourquoi nous créons d'abord les tables.

Il manque le menu déroulant pour le choix du pays. Pour le construire, nous nous connecterons à la table *pays* et afficherons son contenu. Le menu déroulant HTML pour *pays* serait par exemple :

```
<select name="pays">
 <option value="ES" selected>Espagne</option>
 <option value="FR">France</option>…
</select>
```

name est le nom de la variable généré par le moteur PHP tandis que value est sa valeur. Si le code était absent, le nom affiché (Espagne) serait envoyé. L'attribut selected sélectionne le nom qui sera affiché par défaut à l'ouverture de la page.

Sélection par défaut

Avant d'envoyer le formulaire, nous ferons une vérification avec un langage qui rend cette dernière plus simple : Javascript. Par convenance, nous aurons dans tous les menus déroulants une valeur vide par défaut. Cette astuce permet de savoir si l'internaute a choisi volontairement une valeur. Si, par exemple, vous mettez France par défaut, un internaute négligent de Madagascar ou du Québec laissera la valeur par défaut.

Nous allons ajouter une ligne en haut du fichier *inscription_p1.php*, encadrée par les balises PHP :

```
include "commun/connexion.inc.php";
```

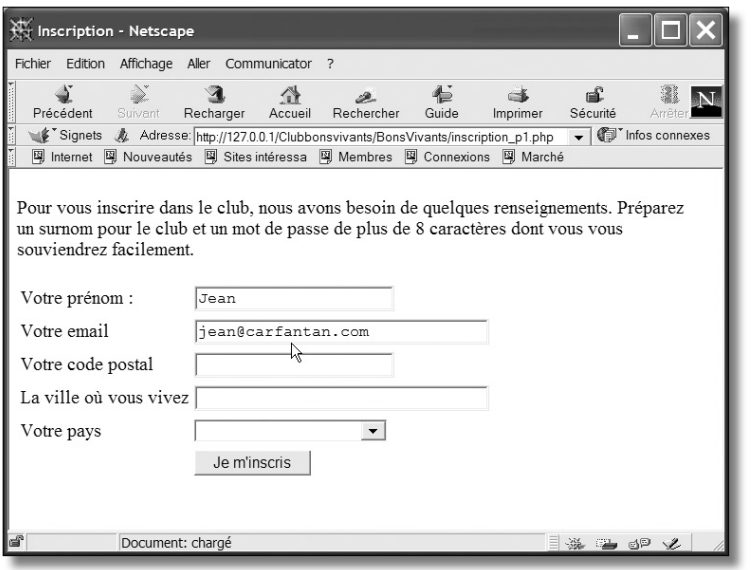
Notre menu en PHP sera :

```
<?
$sql="select * FROM  pays ORDER BY nom";
$resultat=@mysql_query($sql,$id_link);
     echo '<select name="pays">';
     echo '<option value="" selected></option>';
///////////////la première ligne vide///
       while ($rang=mysql_fetch_array($resultat)){
// mysql_fetch_array() crée un tableau avec les lignes du résultat
       $code=$rang['code'];
       $nom=$rang['nom'];
       echo "<option value=\"$code\">$nom</option>";
       }
     echo '</select>';
?>
```

Dans cette deuxième version, nous utilisons des commandes SQL pour récolter l'information. Nous employons la fonction `mysql_query()` qui prend comme paramètres la commande SQL et l'identifiant de connexion. Son résultat est transformé en tableau par la fonction `mysql_fetch_array()`. Une boucle `while` décline ensuite le tableau sous forme de lignes.

Il suffit alors d'inclure les valeurs de champ (en l'occurrence `code` et `nom`) dans un HTML reconstitué avec la commande **echo**. Le nom du champ est aussi la clé ou l'indice du tableau.

Nous créons une valeur vide pour la balise `option` comme sélection par défaut, afin de vérifier si le champ est renseigné par le visiteur. N'oubliez pas, non plus, d'attribuer aux guillemets des signes d'échappement (barre oblique inversée) à l'intérieur des guillemets.



▲ Figure 4.3 : *Formulaire, inscription première page*

Sans ce code nourri dans la table *pays*, vous seriez obligé d'écrire, ligne après ligne, la liste des deux cents et quelques pays dans la page.

Priorités dans la liste

Les visiteurs de votre site viendront sans doute plutôt des pays francophones. Avec la commande **select**, vous pouvez ordonner la liste pour que les visiteurs ne soient pas obligés de faire défiler deux cents noms de pays avant de trouver le leur. Ainsi, vous ajoutez une colonne nommée `ordre`. Dans cette colonne, indiquez `1` pour les pays francophones. Il vous suffit ensuite de moduler le tri avec `ORDER BY` afin d'avoir d'abord, par ordre alphabétique croissant, les pays dont le champ *ordre* est `1` puis les autres, toujours par ordre alphabétique :

```
<?
echo '<select name="pays">';
echo '<option value=""></option>';
$sql="select * from  pays ORDER BY ordre DESC, nom ASC";
//première requête
$resultat=@mysql_query($sql,$id_link);
   while ($rang=mysql_fetch_array($resultat)){
       $code=$rang['code'];
       $nom=$rang['nom'];
       echo "<option value=\"$code\">$nom</option>";
   }
```

Nous pourrions ajouter `WHERE`.

`WHERE` permet de sélectionner les lignes de la table en fonction de la valeur d'un ou de plusieurs champs. Pour plusieurs champs, nous aurions pu utiliser les mots-clés `AND` ou `OR` :

```
SELECT * from pays WHERE code='FRA' AND ordre=1
```

L'exemple est absurde mais permet de bien comprendre la syntaxe.

La deuxième page

Construisons maintenant le programme pour traiter le formulaire.

Les balises PHP

Les balises qui signalent PHP sont `<? ?>`, `<?php ?>` ou comme pour l'ASP, `<% %>`. Cela dépend de vos fichiers de

configuration. En règle générale, si vous utilisez <?php ?>, vous êtes sûr que ça fonctionne.

Voici les premières lignes du programme :

```
<?
include "commun/connexion.inc.php";
$prenom=$_POST['prenom'];
$code_postal=$_POST['code_postal'];
$ville =$_POST['ville'];
$email=$_POST['email'];
$sql="INSERT INTO membres (prenom, code_postal, ville, pays, email)
 VALUES ('$prenom', '$code_postal', '$ville', '$pays', '$email')";
@mysql_query($sql,$id_link);
?>
```

Les variables dans la requête SQL

Dans l'exemple ci-dessus, les variables sont encadrées par des guillemets simples. Nous avons vu précédemment que des variables à l'intérieur de guillemets simples n'étaient pas remplacées par leur valeur et que PHP les prenait littéralement. Ici ce n'est pas le cas car les guillemets simples sont à l'intérieur de guillemets doubles.

Insérez ensuite le code HTML de la deuxième page :

```
<html>
<head>
  <title>Inscription</title>
</head>
<body>
Veuillez maintenant choisir le nom d'usage que vous porterez dans le
club et le mot de passe pour y accéder. Vous pourrez ainsi
rencontrer d'autres membres et modifier votre profil,
voire le supprimer éventuellement.
<form action="traitement_id.php" method="post" name="page_2">
<table cellspacing="2" cellpadding="2" border="0">
<tr>
 <td>Votre nom d'usage</td>
```

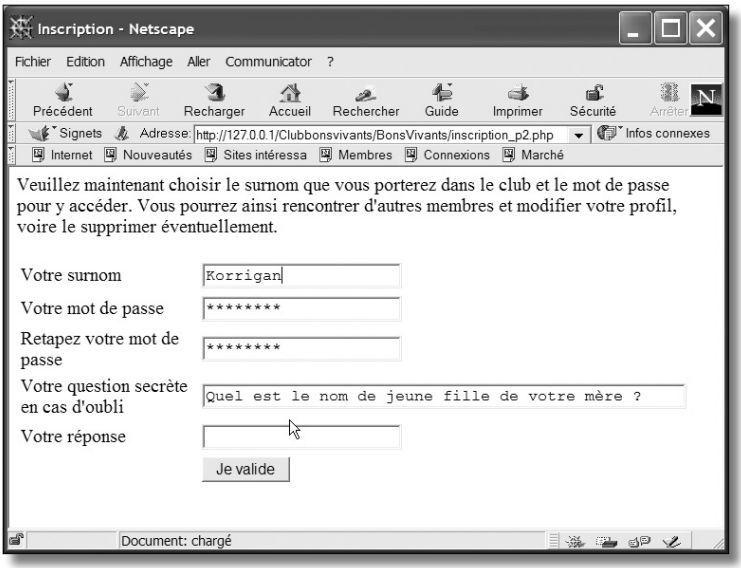
```
 <td><input type="text" name="nom_usage"></td>
</tr>
<tr>
 <td>Votre mot de passe</td>
 <td><input type="password" name="mot_passe"></td>
</tr>
<tr>
 <td>Retapez votre mot de passe</td>
 <td><input type="password" name="mot_passebis"></td>
</tr>
</table>
</form>
</body>
</html>
```

Le champ *input* de type *password* affiche des étoiles à la place des mots. Pour vérifier l'adéquation des deux mots de passe saisis, nous utiliserons le langage Javascript que nous aborderons un peu plus loin dans ce chapitre.

La page est utilisable telle quelle, pourtant nous pouvons encore l'améliorer en ajoutant deux champs : un champ *question* et un champ *réponse*. Ainsi, en cas d'oubli, le membre répondra à une question secrète qui lui permettra de valider son identité avant de recevoir son mot de passe. Nous suggérons une question secrète par défaut, avec l'attribut `value` du champ *question*. Nous ajoutons l'attribut `size` pour que la longueur du champ soit adaptée à la phrase.

```
<tr>
 <td>Votre question secrète en cas d'oubli</td>
 <td><input type="text" name="question" value="Quel est le nom de
jeune fille de votre mère ?" size="50"></td>
</tr>
<tr>
 <td>Votre réponse</td>
 <td><input type="text" name="reponse"></td>
</tr>
```

Il faudra ajouter ces deux colonnes à votre table *membres* avec la commande **ALTER** comme nous l'avons fait précédemment pour ajouter à la table les champs *date_inscription* et *date_misajour*.



▲ Figure 4.4 : *Formulaire, inscription deuxième page*

Nous avons presque fini mais imaginez qu'un internaute paresseux perde son mot de passe et pense qu'il est plus simple de se réinscrire. Il serait alors inscrit deux fois.

Pour éviter cela, il faut effectuer une vérification sur le champ *email* de votre table *membres* :

```
$sql="select clef from membres where email='$email'";
$resultat=@mysql_query($sql,$id_link);
$nombre=mysql_num_rows ($resultat);
if ($nombre >0){
$auth=1;
include "inscription_p1.php";
exit;
}
```

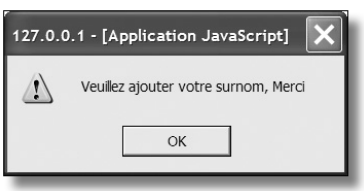
Avant `include "inscription_p1.php";`, il suffit d'ajouter `$auth=1;` et d'insérer une condition dans le formulaire *inscription_p1.php* pour afficher une information :

```
if ($auth==1){
echo "<p><font color=\"#FF0000\"> Vous êtes déjà membre et vous avez
peut-être <a href=\"oubli.php\">oublié votre mot de
passe</a>.</font></p>";
}
```

La troisième page

Pour la troisième page, nous allons insérer le nom d'usage, le mot de passe et la date d'inscription, remercier le nouveau membre du club et lui envoyer un courriel de confirmation. Enfin nous détruirons les entrées non valides dans la table :

```
<?
include "commun/connexion.inc.php";
/*//////////////////DATE SOUS LA FORME AAAAMMJJ//////////////////*/
$moment=date ("Ymd", mktime (0,0,0,date("m"),date("d"),date("Y")));
$nom_usage =$_POST['nom_usage'];
$mot_passe =$_POST['mot_passe'];
$question =$_POST['question'];
$reponse= strtolower($_POST['reponse']);
$sql="INSERT INTO membres (nom_usage, mot_passe, date_inscription,
 question, reponse) VALUES ('$nom_usage', '$mot_passe', '$moment',
 '$question', '$reponse')";
@mysql_query($sql,$id_link);
?>
<html>
<head>
  <title>Inscription</title>
</head>
<body>
Merci de votre inscription et bienvenue dans le Club des Bons
Vivants. Nous venons de vous envoyer un courriel de confirmation
avec votre nom d'usage et votre mot de passe.
<CENTER><i>Le webmestre</i></CENTER>
</body>
</html>
```



◄ Figure 4.5 : *Formulaire, remerciement*

Nous avons presque fini mais imaginez que deux internautes choisissent le même nom d'usage. Cela invaliderait les procédures de votre club.

Pour éviter cela, il faut effectuer une vérification dans votre table *membres* :

```
$sql="SELECT clef FROM membres WHERE nom_usage='$nom_usage'";
$resultat=@mysql_query($sql,$id_link);
$nombre=mysql_num_rows ($resultat);
if ($nombre >0){
$auth=2;
include "inscription_p2.php";
exit;
}
```

Le reste du code ne sera pas exécuté grâce à la commande **exit**. Ensuite, il suffit d'insérer une condition dans le formulaire *inscription_p2.php* pour afficher une information :

```
if ($auth==2){
echo "<p><font color=\"#FF0000\">Votre nom d'usage existe déjà dans
la base de données. Soit vous êtes déjà membre et vous avez
<a href=\"oubli.php\">oublié votre mot de passe</a>. Soit il s'agit
d'une simple coïncidence, dans lequel cas, entrez un nouveau nom
d'usage...</font></p>";
}
```

Enfin, dans la troisième page, nous ajoutons :

```
$hier=date ("Ymd", mktime (0,0,0,date("m"),date("d")-1,date("Y")));
$sql="DELETE FROM membres WHERE nom_usage='' AND date_misajour<$hier";
@mysql_query($sql,$id_link);
}
```

Nous profitons de ce programme pour ajouter ces quelques lignes de code, enlevant ainsi les enregistrements incomplets dont la création date de plus d'un jour et nettoyer la table.

Fabriquons maintenant puis envoyons le courriel de bienvenue.

La fonction mail()

Le courriel est envoyé avec la fonction `mail()`. Il peut être expédié sous plusieurs formats, le format texte ou le format *HTML*. Le format *HTML*, mieux présenté, n'est pas accepté par tous les logiciels de courrier.

mail()

Cette fonction permet d'utiliser le serveur SMTP, c'est-à-dire le serveur d'envoi de courriel

Syntaxe `mail (string adresse, string sujet, string message)`

Options `[, string en-têtes]`

Le résultat retourné est TRUE ou FALSE.

Nous allons d'abord placer le contenu dans une variable :

```
$contenu="Bienvenue $prenom,\nVous faites désormais partie du club
des Bons Vivants. Votre nom d'usage dans le Club est $nom_usage et
votre mot de passe $mot_passe. Conservez-le afin d'avoir accès au
Club.\nCordialement\nle Webmestre, François Rabelais.\n";
```

Format texte

```
$entete="From: \"le club des Bon Vivants\"
 <contact@bons-vivants.org>\n";
mail ("$email","BIENVENUE AU CLUB","$contenu",$entete);
```

Format HTML

```
$contenu="Bienvenue $prenom,\nVous faites désormais partie du Club
 des Bons Vivants. Votre nom d'usage dans le Club est
 <B>$nom_usage</B>
 et votre mot de passe <B> $mot_passe</B>. Conservez-le
 afin d'avoir accès au Club.\nCordialement\nle Webmestre, François
 Rabelais.\n";
//nous créons le message
$contenu=nl2br($contenu);
//nous créons les paragraphes en HTML avec la fonction nl2br()
//nous composons le courriel
$message="<HTML>\n";
$message.="<HEAD>\n";
$message.="<META NAME=\"GENERATOR\"
 Content=\"Microsoft DHTML Editing Control\">\n";
$message.="<TITLE></TITLE>\n";
// vous pouvez y mettre un titre si vous voulez
$message.="</HEAD>\n";
$message.="<BODY><FONT face=Arial><FONT color=#000000>\n";
```

```
$message.="<P><div align=\"center\"><STRONG><FONT size=4>
Bienvenue $prenom dans le club des Bons Vivants</FONT></STRONG>
</div></p></FONT> </P><FONT color=#000000>
<font face=\"Arial, Helvetica, sans-serif\" size=\"2\">\n";
$message.=$contenu;
$message.="</FONT></P>\n";
$message.="</BODY>\n";
$message.="</HTML>\n";
```

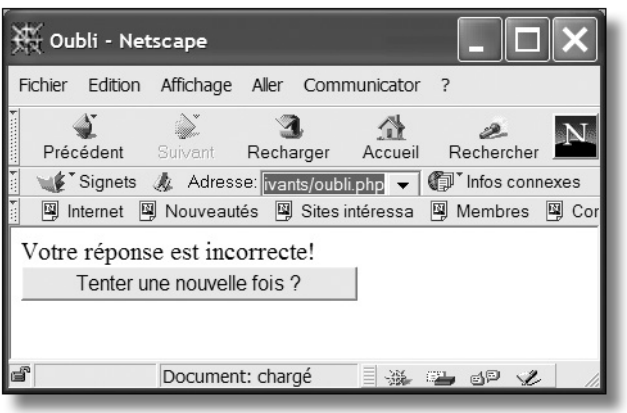
Composons maintenant les en-têtes du message :

```
$entetes = "From: \"le club des Bon Vivants\"
 <contact@bons-vivants.org>\n";
$entetes .= "X-Sender: <contact@bons-vivants.com>\n";
$entetes .= "X-Mailer: PHP\n"; // mailer
$entetes .= "X-Priority: 1\n";
$entetes .= "Return-Path: <contact@bons-vivants.org>\n";
$entetes .= "Content-Type: text/html; charset=iso-8859-1\n";
```

et envoyons le tout :

```
mail ("$email","BIENVENUE AU CLUB","$message",$entetes);
```

Pour ajouter des
 à chaque \n, nous utilisons la fonction `nl2br()` que vous connaissez.



▲ Figure 4.6 : *Courriel au format HTML*

La fonction mail()

Avant de choisir un hébergeur, assurez-vous que la fonction `mail()` est disponible sur le serveur.

4.10 Les fonctions du langage PHP pour MySQL

Le langage PHP tire profit de la simplicité de son interface avec les bases de données, en particulier *MySQL*. Voici les fonctions que vous utiliserez le plus souvent.

Tab. 4.4 : Les principales fonctions d'accès à une base MySQL

Fonction	Description
`mysql_connect()`	Ouvre une connexion.
`mysql_create_db()`	Crée une base.
`mysql_query()`	Exécute une requête et retourne un identificateur de résultats ou `FALSE`.
`mysql_fetch_array()`	Distribue un résultat dans un tableau à index numérique ou un tableau de hachage.
`mysql_num_rows()`	Retourne le nombre de rangs à partir d'un résultat.

4.11 La vérification des formulaires

Dans un site interactif, vous devez composer avec les erreurs de frappe, les problèmes de compréhension et le "sabotage". Ainsi, la vérification des champs obligatoires est indispensable. Il existe deux façons de vérifier un formulaire :

- la vérification instantanée avec le langage Javascript ;
- la vérification pendant le traitement de l'information avec PHP.

La vérification avec Javascript

Javascript est un langage côté client, c'est-à-dire qu'il est interprété directement par le navigateur web. L'intérêt de cette vérification est qu'elle oblige le visiteur à corriger le champ erroné, donc le serveur envoie uniquement des informations valides dans le programme de traitement. C'est l'approche pour la correction d'erreurs éventuelles. Nous n'emploierons PHP que pour des champs qui posent problème à Javascript, comme le menu à choix multiple ou tout champ qui comprend plusieurs éléments à sélectionner.

Nous ne verrons dans cette partie que la syntaxe Javascript utile dans le cas d'un formulaire. Pour approfondir, nous vous donnerons les URL de sites intéressants en annexe.

Quelques notions indispensables

La syntaxe de ce langage est légèrement différente de celle de PHP, même si vous disposez aussi des variables, des tests, des boucles, des fonctions (celles qui préexistent et celles que vous créez vous-même) et des objets (thème que nous n'évoquerons pas dans ce livre).

Les fonctions

Les fonctions `Javascript` sont placées dans l'en-tête HTML tandis qu'elles sont appelées à partir de la balise `FORM` ou d'un champ du formulaire (y compris le bouton **SUBMIT**). Ce langage vérifie directement, dans la mémoire vive du navigateur, les données entrées par l'utilisateur. Une fonction Javascript est appelée par un événement. Dans un formulaire, cet événement est un changement intervenu dans un champ ou un clic sur le bouton **SUBMIT**. Comme dans le langage PHP, une fonction peut en appeler une autre.

Variables et objets

Une variable doit être déclarée. Son nom n'est pas précédé d'un `$`.

Les objets comme les éléments d'une page web sont définis dans une hiérarchie arborescente nommée le DOM (`Document Object Model`). `FORM` est un de ces objets. La notation pour définir le premier champ, *nom_usage*, dans la deuxième page est `window.document.forms[0].nom_usage`. Si vous n'avez qu'une fenêtre, c'est-à-dire que vous n'avez

pas ouvert de nouvelle fenêtre avec une fonction Javascript, vous pouvez élider le mot `window` et écrire `document.forms[0].nom_usage`. Javascript crée automatiquement un tableau d'objets, ainsi le même objet `nom_usage` peut être défini à l'aide des tableaux et s'écrire indifféremment :

- `document.forms[0].nom_usage` ;
- `document.page_2.elements[0]` ;
- `document. forms[0].elements[0]`.

Tableaux

Le premier élément d'un tableau a un indice `0` comme en PHP et le nom des tableaux est au pluriel, même si vous n'avez qu'un formulaire ou qu'un élément de formulaire.

Un objet `String` dispose de la propriété `length` que nous utiliserons pour vérifier que le champ a été rempli correctement. L'objet `window` contient la méthode `alert()` qui permet d'afficher une fenêtre d'avertissement avec un bouton OK. Étudions plus précisément maintenant les objets du formulaire et leurs propriétés.

Les objets de formulaire

Les champs texte ont comme propriété `name`, `defaultValue` et `value`.

Ainsi l'expression `document.forms[0].nom_usage.value.length` renferme la longueur de la valeur que l'internaute entre dans le champ *nom_usage*. Si cette valeur est égale à 0, on peut penser que l'internaute n'a rien entré dans ce champ *nom_usage*.

Si vous voulez lire un chiffre comme valeur numérique dans un champ texte, vous devrez utiliser la fonction `parseInt()`. Si nous posions la question du nombre de membres vivant au foyer, pour vérifier le nombre il faudrait écrire : `nb_personnes=parseInt(document.forms[0].nb_foyer.value)`. La valeur devra être d'au moins une personne et pour utiliser "plus grand que", la méthode `parseInt()` est nécessaire afin de transformer la chaîne de caractères en un entier.

Tab. 4.5 : Quelques méthodes Javascript liées aux formulaires

Méthodes .	Description
`focus()`	Met le curseur dans le champ.
`blur()`	Désactive le champ en sortant le curseur.
`select()`	Sélectionne le contenu du champ.
`alert()`	Crée une fenêtre de validation qui présente un bouton OK.
`prompt()`	Crée une fenêtre de saisie qui demande une réponse à une question.
`confirm()`	Crée une fenêtre de confirmation qui présente un bouton OK et un bouton **annuler**.
`parseInt()`	Récupère les chiffres sous forme numérique dans un champ *text*.

Les cases ou les boutons à cocher

Une case à cocher est un élément individuel à cocher. Il est individuel dans la mesure où un nom le différencie des autres cases.

Un bouton **radio** est un bouton solidaire de son groupe puisque chaque bouton **radio** d'un groupe de boutons possède le même nom. Le choix est un choix exclusif. Si vous cochez un bouton **radio**, les autres se mettront dans l'état vide.

Menus déroulants à choix unique

Le choix de l'internaute dans un menu déroulant est stocké dans la propriété `selectedIndex`. Ensuite, le résultat nous permet de connaître la valeur choisie :

```
index_pays=document.forms[0].pays.selectedIndex;
lepays= document.forms[0].pays.options[index_pays].value;
```

Ici la valeur est cherchée dans le tableau d'options avec l'index désigné par la propriété `selectedIndex`.

Quelques références complémentaires

Pour compléter cette étude du langage Javascript afin de vérifier vos formulaires, il vous reste à connaître les paramètres des objets présents dans le tableau `elements[]`.

Tab. 4.6 : Les paramètres Javascript pour les champs de formulaire

Propriétés	Champs	Types
`name`	Tous	`String`
`value`	Tous	`String`
`checked`	Case à cocher, bouton radio	Booléen
`selectedIndex`	Menu déroulant	`Int`
`length`	Bouton radio, menu déroulant (nombre d'éléments)	`Int`

Voici encore des événements qui peuvent déclencher l'appel à une fonction.

Tab. 4.7 : Événements Javascript

Événements	Description
`onSubmit`	L'internaute clique sur le bouton **submit**.
`onClick`	L'internaute clique sur un lien, un bouton radio, une case à cocher…
`onSelect`	L'internaute sélectionne une partie ou tout le texte d'un champ texte.
`onFocus`	L'internaute clique dans un champ texte.
`onChange`	L'internaute change le contenu d'un champ et en sort.
`onBlur`	L'internaute sort d'un champ texte.

Voyons maintenant les scripts Javascript à ajouter à nos pages de formulaires.

Les scripts Javascript

Ces scripts consistent en une ou plusieurs fonctions. Elles sont lancées par un événement `onSubmit` qui intervient quand l'internaute clique sur le bouton **submit**.

Première page

Les fonctions seront appelées à partir de la balise `FORM` :

```
<form action="inscription_p2" method="post" onSubmit="return
envoi()">
```

L'événement `onSubmit` ne pourra être opérant que s'il reçoit la valeur `TRUE` de la fonction `envoi()`.

Nous allons d'abord créer une fonction `envoi()`pour empêcher l'internaute de cliquer plusieurs fois sur le bouton **submit**.

```
var unique=0;
//ici la variable est déclarée avec ou sans le mot var
function envoi() {
if (unique == 0){
unique++;
return true;
}
else {
alert("Envoi en cours...!");
return false;
}
}
```

Nous déclarons d'abord une variable globale `unique` (située dans la page, en dehors du bloc d'exécution de la fonction). Elle sera initialisée à 0 au moment du chargement de la page dans le navigateur. Ensuite, quand l'internaute cliquera sur le bouton **SUBMIT**, il incrémentera la valeur à 1. Ainsi, si l'internaute clique une deuxième fois sur le bouton, une alerte s'affichera et ce deuxième envoi sera bloqué. Cette fonction évite d'envoyer les données deux fois. L'internaute perd patience quand le serveur est encombré et que la deuxième page est longue à s'afficher.

Maintenant que vous avez compris le principe, créons les autres fonctions : `verif_champs()` et `verif_email()`. Comme son nom l'indique, la première fonction vérifie que les champs du formulaire sont renseignés,

tandis que la deuxième vérifie que le texte entré est pourvu d'un arobase qui caractérise une adresse e-mail. Cette dernière fonction est peu sophistiquée mais elle suffit car nous relevons les erreurs involontaires. Il est difficile de déceler une fausse adresse e-mail dans la mesure où elle est bien formatée. Le formulaire de la première page contient cinq champs dont quatre champs *text*. Le dernier est un menu.

Pour vérifier que du texte a été entré, nous utilisons la propriété `length` qui donne la longueur de l'objet `String`.

La fonction `verif_champs()` :

```
function verif_champs(){
if (document.forms[0].prenom.value.length<1){
   alert("Veuillez ajouter votre prénom, Merci");
   document.forms[0].prenom.focus();
   return false;}
if (document.forms[0].ville.value.length<1){
   alert("Veuillez ajouter votre ville, Merci");
   document.forms[0].ville.focus();
   return false;}
if (document.forms[0].code_postal.value.length<1){
   alert("Veuillez ajouter votre code postal, Merci");
   document.forms[0].code_postal.focus();
   return false;}
index_pays=document.forms[0].pays.selectedIndex;
lepays= document.forms[0].pays.options[index_pays].value;
if (lepays==0){
   alert("Veuillez définir votre pays, Merci");
   return false;}
if (document.forms[0].email.value.length<5){
   alert("Veuillez ajouter votre email, Merci");
   document.forms[0].email.focus();
   return false;}
return true;
}
```

La fonction `verif_email ()` :

```
function verif_email (){
var c=document.forms[0].email.value;
var test="" + c;
for(var k = 0; k < test.length;k++)
{
   var d = test.substring(k,k+1);
   if(d == "@")
```

```
   {
      return true;
   }
}
alert("Votre E-mail n'est pas valide, Merci");
document.forms[0].email.focus();
return false;
}
```

La fonction `envoi()` :

```
function envoi(){
if (unique == 0){
unique++;
return true;
}
else {
alert("Envoi en cours...!");
return false;
}
}
```

À chaque fois que la fonction envoie un `return`, son exécution s'arrête. C'est pourquoi nous avons placé l'expression `return true` après tous les tests. Chaque test faux arrête la chaîne d'exécution. Pour que le formulaire soit envoyé, il faut qu'il reçoive un `return true`. Nous allons donc créer une fonction maître regroupant les trois fonctions, qui s'appellera `verif_formulaire()`. Cette fonction retournera `true` seulement si les trois fonctions sont vraies.

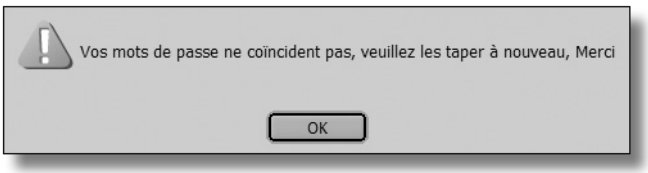
```
function verif_formulaire(){
if (verif_champs()==true && verif_email()==true && envoi()==true){
return true;
}
return false;
}
```

Le script complet est sur le site. Il vous suffit maintenant d'appliquer ce que vous venez d'apprendre sur la deuxième page. Vous aurez néanmoins besoin de faire un test supplémentaire sur les deux champs du mot de passe et sur la longueur du mot de passe qui doit dépasser sept caractères :

```
if (document.forms[0].mot_passe.value.length<8){
 alert("Veuillez ajouter votre mot de passe de
 plus de 7 caractères, Merci");
```

```
 document.forms[0].mot_passe.focus();
 return false;
}
if (document.forms[0].mot_passe.value!=
 document.forms[0].mot_passebis.value){
 alert("Vos mots de passe ne coïncident pas,
 veuillez les taper à nouveau, Merci");
 document.forms[0].mot_passe.focus();
 return false;
}
```

La méthode `focus()` placera le curseur dans le champ *mot_passe* en cas d'erreur.



▲ Figure 4.7 : *Vérification en langage Javascript*

Javascript est très pratique et assez simple à utiliser. Il est des cas, cependant, où il n'est pas possible de s'en servir.

Le code PHP

Nous ne pouvons utiliser Javascript pour vérifier des champs qui envoient un tableau dans le cas des groupes de cases à cocher et des menus déroulants à choix multiple. Pour que le programme PHP, qui traite le formulaire, puisse prendre en compte un de ces champs, il nous faut ajouter des crochets dans le nom, ce que Javascript ne reconnaîtra pas comme un nom de champ.

Le formulaire avec une sélection multiple

Reprenons l'exemple de `pays` qui permettrait plusieurs sélections si nous posions la question de la nationalité et non celle du pays. Ainsi le candidat pourrait choisir plusieurs options en cas de double nationalité :

```
<tr>
 <td>Votre nationalité (<i>Vous pouvez sélectionner plusieurs pays
en appuyant sur la touche "ctrl" tout en cliquant sur la sélection
</i>)</td>
 <td><input type="text" name="pays[]" size="1" multiple></td>
</tr>
<?
echo '<select name="pays">';
echo '<option value="" selected></option>';
$sql="select * from  pays";
$resultat=@mysql_query($sql,$id_link);
   while ($rang=mysql_fetch_array($resultat)){
       $code=$rang['code'];
       $nom=$rang['nom'];
       echo "<option value=\"$code\">$nom</option>";
   }
echo '</select>';
?>
```

Le traitement du formulaire

Ici les valeurs du tableau sont de longueurs égales (trois caractères), ce qui en simplifie le traitement :

```
<?
include "commun/connexion.inc.php";
$pays=$_POST['pays'];
if (is_array($pays)){
foreach (pays as $clef=>$valeur){
$touslespays.=$valeur;
}
$pays=$touslespays;
$sql="INSERT INTO membres (prenom, code_postal, ville, pays, email)
 VALUES ('$prenom', '$code_postal', '$ville', '$pays', '$email')";
@mysql_query($sql,$id_link);
?>
```

Les différents éléments ont été concaténés. Chaque membre peut avoir trois nationalités et chaque pays possède un code de trois caractères. Il vous faut élargir la colonne à douze caractères pour conserver un peu de marge.

Ensuite, il sera facile de récupérer les données dans un tableau avec un `substr()` :

```
for($i=0;$i<strlen($pays);$i=3){
$code_pays[]=substr($pays,$i,3);
}
```

Maintenant, nous allons vérifier si une ou plusieurs options ont été sélectionnées dans le champ *pays[]* :

```
if (!$pays[]){
include "inscription_p1.php";
exit;
}
```

Si aucune option n'a été sélectionnée, la variable `$pays[]` est vide, c'est-à-dire qu'elle a été transmise par le programme mais que sa valeur est vide.

isset() et empty()

Dans le cas de `empty()`, la valeur a été initialisée mais elle ne contient pas de valeur, alors que la fonction `isset()` interroge la variable pour savoir si elle a été initialisée, et donc si elle existe.

À l'intérieur de ce bloc de test, nous utilisons la commande **include**. Ainsi le contenu de la première page va s'afficher à l'intérieur de la page courante. La commande **exit** est placée afin d'éviter que la page 2 n'apparaisse à la suite de la page 1.

include

Dans le champ du navigateur apparaîtra l'URL de la page 2 car ici, la page 1 est seulement incluse. L'internaute a besoin d'une explication sur le retour à cette première page. Nous pouvons afficher une annonce dans un bloc de test contenu dans la première page. Il suffit alors de l'afficher si la variable Apache `$REQUEST_URI` ne correspond pas à la page du script (en l'occurrence, cette variable donnera l'adresse de la page 2 et non de la page 1) :

```
If (ereg($REQUEST_URI, 'inscription_p2.php'){
echo ' <font color="#FF0000">Veuillez sélectionner au moins
 un pays, Merci.</font>';
}
```

Ici l'annonce apparaîtra en rouge si la page a été appelée par une commande **include** lancée de la page 2.

Le visiteur n'a encore inséré aucune donnée et il est revenu à son point de départ. Il lui faudra renseigner à nouveau tous les champs. Nous allons donc insérer la valeur qu'il avait entrée dans chaque champ afin de lui éviter cette corvée. D'autre part, dans d'autres circonstances, cette opération permettra à l'internaute, comme nous allons le voir plus loin, de modifier lui-même ses coordonnées en cas de changement d'adresse.

Nous afficherons chaque champ par un `echo`, de façon à afficher la valeur de la variable qui correspond à chaque champ.

Revoyons les quatre premiers champs de la page 1 :

```
<?
if (ereg($_SERVER['REQUEST_URI'], 'inscription_p2.php'){
echo '<p><font color="#FF0000">Veuillez sélectionner au moins
 un pays, Merci.</font></p>';
}
?>
<form action="inscription_p2.php" method="post"
 onSubmit="return verif_formulaire()">
<table cellspacing="2" cellpadding="2" border="0">
<tr>
 <td>Votre prénom :</td>
 <td>
 <?
 echo "<input type=\"text\" name=\"prenom\" value=\"$prenom\">";
 ?>
 </td>
</tr>
```

Nous faisons la même manipulation sur le reste des champs du formulaire. Nous incluons ici la variable qui nous a été envoyée par chaque champ au moment où l'internaute a soumis le formulaire. Nous devons ajouter des caractères d'échappement à chaque guillemet. Ainsi il n'aura plus besoin d'entrer les mêmes renseignements. D'autre part, nous pourrons utiliser cette facilité quand l'internaute voudra modifier ses données personnelles. Pour anticiper cette modification, nous allons opérer de la même façon sur le champ *pays*.

Une option dans un menu déroulant est sélectionnée par défaut par l'attribut `SELECTED` dans sa balise, il nous suffit donc de modifier légèrement le code pour l'intégrer dans le menu déroulant **pays** :

```
<?
echo '<select name="pays">';
echo '<option value=""></option>';
$sql="select * from  pays ORDER BY ordre DESC, nom ASC";
$resultat=@mysql_query($sql,$id_link);
   while ($rang=mysql_fetch_array($resultat)){
       $code=$rang['code'];
       $nom=$rang['nom'];
       echo "<option value=\"$code\"";
if ($pays==$code){
echo " SELECTED";
}
echo ">$nom</option>";
   }
```

Dans cet exemple, nous avons ajouté un test pour savoir si la variable `code` courante correspondait à la variable `pays`. Si le cas se présente, nous ajoutons `SELECTED`. Notre but est d'afficher l'option précédemment choisie.

```
<option value="BEN" SELECTED>Bénin</option>
```

L'internaute ne pourra valider ses données tant que le format ne sera pas conforme.

Variables prédéfinies liées au chemin ou à l'URL

Ces variables sont utilisées pour faire une application qui fonctionne sur tout serveur, puisque ces noms se modifient suivant l'URL. Elles permettent donc de créer des applications portables.

Voyez l'application *Écoute de fichiers* dans le chapitre *Moteurs de recherche et expressions rationnelles*.

Tab. 4.8 : Les variables prédéfinies

Variable	Description
`$DOCUMENT_ROOT`	Chemin à partir de la racine des documents
`$PATH_TRANSLATED`	Chemin local propre au serveur

Tab. 4.8 : Les variables prédéfinies

Variable	Description
`$QUERY_STRING`	Données qui suivent le point d'interrogation dans l'URL, passées avec la méthode `GET`
`$REQUEST_URI`	Adresse affichée dans le champ location du navigateur
`$SCRIPT_FILENAME`	Chemin complet du fichier
`$SCRIPT_NAME`	Chemin du fichier relatif au répertoire *www* ou *htdocs*

4.12 Tout sur un seul fichier : oubli.php

Nous allons maintenant préparer un formulaire qui apparaîtra dans une fenêtre en cas d'oubli du mot de passe. C'est un simple formulaire avec deux champs : *nom d'usage* et *réponse*, précédé de la question, puis vient la page avec l'annonce qui vient d'être envoyée par courriel. Tous ces traitements seront réunis sur un fichier.

L'information est reçue par courriel pour authentifier l'adresse e-mail. Nous verrons qu'il est intéressant aussi d'envoyer régulièrement une lettre aux membres et d'y inclure le nom d'usage avec le mot de passe afin de le leur rappeler.

La réflexion

Le formulaire peut se rappeler lui-même avec la variable `$PHP_SELF`. Nous pouvons également utiliser un champ caché *passage* pour marquer chaque étape à l'intérieur d'une variable. Les champs cachés posent un problème de sécurité car il suffit qu'un internaute récupère la source du formulaire, modifie ce dernier et le soumette pour créer une faille dans la sécurité. Il vaut mieux l'utiliser uniquement pour des marqueurs ou pour passer des numéros d'identification complexes, impossibles à inventer. Ici nous l'employons comme marqueur et sa modification ne permet à l'internaute de récupérer aucun mot de passe.

La plate-forme commune

Voici la plate-forme commune pour ce fichier :

```
<?
include "commun/connexion.inc.php";
?>
<html>
<head>
<title>Oubli</title>
<meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1"></head>
<body bgcolor="#FFFFFF" text="#000000">
<?
echo "<form name=\"form1\" method=\"post\" action=\"$PHP_SELF\">";
?>
</form>
</body>
</html>
```

Tous les autres éléments sont optionnels. Ils dépendent des étapes suivies par l'internaute.

Les variables comme marqueurs

À chaque étape, un champ caché dans le formulaire décrira l'étape suivante. La première étape est sans marqueur. À l'étape 2, nous entreposons la variable `$nom_usage`, nécessaire pour interroger ensuite la table *membres* sur la réponse à la question. Enfin si la réponse est vraie, un bouton apparaît pour fermer la fenêtre. Si la réponse est fausse, l'internaute se voit proposer une nouvelle tentative. Dans ce dernier cas, la variable `passage` contient le chiffre 3, nous la réduisons à 0 pour recommencer le processus jusqu'à 3 fois. Pour limiter les tentatives à 3, nous stockerons une variable `tentative` dans une session. Au bout de trois, nous proposerons à l'internaute d'envoyer un courriel au webmestre qui jugera bon ou non de lui envoyer son nom d'usage et son mot de passe. C'est pourquoi les champs du formulaire (dont le bouton **Soumettre**), les requêtes SQL et l'affichage en général dépendront du déroulement de chaque étape.

Les sessions

Si dans une page, vous avez besoin d'une variable de session, il faut la réactiver avec la fonction `session_start()` placée en haut de la page. Cette fonction crée une session ou réveille la session courante si elle existe. Si vous utilisez la fonction `session_register()`, la fonction `session_start()` est implicite, vous n'avez donc pas à la rajouter.

Premier passage : $passage==0

La première étape repose sur la non existence de la variable `passage` :

```
If (!$passage){
```

Il est possible d'utiliser `not` qui décrit une variable non initialisée, ou contient une valeur vide ou une valeur numérique 0.

Nous affichons un champ *nom_usage*. Dans le formulaire, nous ajoutons :

```
$passage=$_GET['passage'];
if (!$passage){
echo ' Votre nom d'usage : <input type="text" name="nom_usage">';
echo ' <input type="hidden" name="passage" value="1">';
echo '<BR><input type="submit" name="Submit" value="Soumettre">';
}
```

Nous avons ajouté un champ caché qui initialisera la variable passage à 1 pour l'étape suivante lors de l'envoi du formulaire.

Deuxième passage : $passage==1

Nous testons la valeur de la variable `passage`. Auparavant, nous allons conserver le nom d'usage du membre pour créer les requêtes suivantes afin de rechercher la question puis comparer les réponses et lui envoyer une réponse par courriel à la troisième étape.

```
session_register("nom_usage");
if ($passage==1){
$sql="select question from membres where nom_usage='$nom_usage'";
$resultat=@mysql_query($sql,$id_link);
$rang=mysql_fetch_array($resultat);
```

```
$question=$rang['question'];
}
```

Nous affichons ensuite la question et le champ *reponse* :

```
elseif ($passage==1){
///suite au if (!$passage) mis plus haut////
echo $question;
echo "<BR>\n";
echo ' Votre réponse : <input type="text" name="reponse">';
echo '<input type="hidden" name="passage" value="2">';
echo '<BR><input type="submit" name="Submit" value="Soumettre">';
}
```

Les variables de session

Une session est l'unité de temps pendant laquelle l'internaute visite des pages sur un site. Les variables entreposées dans une session conservent leur valeur au fil des pages. Pour entrer une variable dans une session, il suffit d'utiliser la fonction `session_register()`. Attention, elle ne prend pas la variable comme paramètre mais son nom, ici `nom_usage`.

Troisième passage : $passage==2

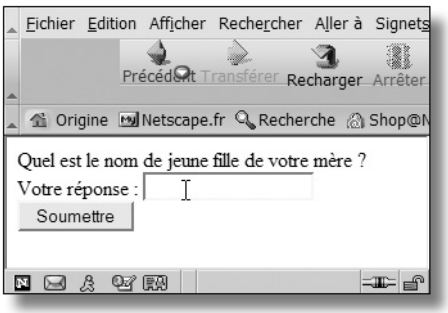
Nous testons la présence de la variable `$reponse` :

```
if ($passage==2){
$sql="select nom_usage, mot_passe, reponse from membres
 where nom_usage='$nom_usage' ";
$resultat=@mysql_query($sql,$id_link);
$rang=mysql_fetch_array($resultat);
$reponse_vraie=$rang['reponse'];
$mot_passe =$rang['mot_passe'];
$reponse =strtolower($($reponse);
}
```

Puis nous affichons le résultat :

```
if ($passage==2){
if ($reponse==$reponse_vraie){
echo "Votre réponse est exacte, nous venons de vous
 envoyer un email avec votre mot de passe";
```

```
echo "<BR>\n";
echo "<input type=\"submit\" name=\"Submit\"
 value=\"D'accord!\" onClick=\"window.close()\">";
$message="Bonjour";
}
else {
echo "Votre réponse est incorrecte!</a>";
echo '<input type="hidden" name="passage" value="3">';
echo "<BR>\n";
echo "<input type=\"submit\" name=\"Submit\"
 value=\"Tenter une nouvelle fois ?\">";
}
}
```



◄ Figure 4.8 : *La troisième étape de l'application, oubli*

Vous avez sans doute remarqué dans la première partie de cette étape que nous n'avons pas utilisé `while`. Effectivement, ce n'est pas nécessaire car nous recherchons une seule ligne.

Les tentatives $passage==3

Pour la deuxième tentative, nous vidons la variable `passage` pour revenir au début de la première étape. Nous allons limiter le nombre de tentatives en enregistrant la variable `$tentative` dans la session courante :

```
if ($passage==3){
$passage="";
if (!$tentative){
$tentative=1;
}
else{
$tentative++;
```

```
}
session_register(tentative);
}
```

Ainsi quand l'internaute atteint la fin de la troisième tentative avec une réponse fausse à la question, un nouvel écran lui demande d'envoyer un courriel au webmestre.

L'ouverture d'une fenêtre avec une fonction Javascript

Le formulaire *oubli.php* n'a pas besoin de tout l'écran pour s'afficher, nous allons donc créer une nouvelle fenêtre avec une fonction Javascript que nous appellerons `ouvrir_vasistas()` :

```
<SCRIPT LANGUAGE="JavaScript">
<!--
function ouvrir_vasistas(adresse, nom_fenêtre,caracteristiques) {
  window.open(adresse, nom_fenêtre,caracteristiques);
}
</SCRIPT>
```

Quittez la page *oubli.php* et dans votre page d'accueil, ajoutez le lien :

```
Vous êtes déjà membre et vous avez peut-être
<a href="#"
onClick="ouvrir_vasistas('oubli.php','vasistas','menubar=no,
scrollbars=no,statusbar=no, width=300,height=200')">
oublié votre mot de passe</A>.
```

L'application est alors bouclée.

4.13 Récapitulation de la syntaxe des requêtes

Au cours de ce chapitre, l'apprentissage de la syntaxe SQL et des fonctions MySQL a été dispensé au fur et à mesure des applications. Récapitulons-les.

Voici d'abord les principales commandes SQL. Ce sont des commandes communes à la plupart des bases de données. Seule leur syntaxe peut présenter des petites particularités (les types de guillemets…).

Tab. 4.9 : Les principales commandes SQL

Commande	Description	Syntaxe
CREATE	Crée une table	CREATE TABLE *nom_table* (*nom_colonne type*(*longueur*)…)
DELETE	Efface des données	DELETE FROM *nom_table*
INSERT	Insère des données	INSERT INTO *nom_table* [(*colonne1*,…)] VALUES (*valeur_colonne1*,…)
SELECT	Extrait des données	SELECT *nom_colonne1*… FROM *nom_table*
UPDATE	Modifie des données	UPDATE *nom_table* SET *nom_colonne1='nouvelle_valeur'*,…

Les fonctions de *MySQL* présentent l'inconvénient de consommer de la ressource serveur, surtout si votre base est bien remplie. Mais elles permettent de vous épargner des lignes de codes en langage PHP.

Tab. 4.10 : Quelques fonctions MySQL

Fonction	Description	Syntaxe
AVG()	Retourne la moyenne des différents nombres	AVG(nom_colonne1)
COUNT()	Compte le nombre de lignes	COUNT (*)
RAND()	Permet d'effectuer un tri aléatoire avec ORDER BY	ORDER BY RAND ()

Voici un certain nombre de clauses SQL qui modulent le résultat :

WHERE

Détermine les données sur lesquelles s'appliquent les commandes SQL DELETE, SELECT, UPDATE

Syntaxe : `WHERE nom_colonne='valeur'`

Options AND, OR, NOT

Exemple `DELETE FROM nom_table where nom_colonne1=2 OR nom_colonne1=5`

Explication AND : les deux expressions sont vraies, OR : une expression est vraie, NOT : une expression est fausse. Ces trois opérateurs peuvent se combiner avec des parenthèses

Prenons par exemple :

```
SELECT * from membres where (pays='HTI' OR pays='DMA')
AND date_misajour BETWEEN 20020401 AND 20020430.
```

Les parenthèses donnent des priorités, ainsi OR est traité avant AND.

DISTINCT

Regroupe les données identiques

Syntaxe : `DISTINCT nom_colonne`

Options sans

Exemple `SELECT DISTINCT nom_colonne1 FROM nom_table`

Explication retire les doublons du résultat

GROUP BY

Regroupe les données identiques

Syntaxe : `GROUP BY nom_colonne1, nom_colonne2…`

Options sans

Exemple `SELECT * FROM nom_table where nom_colonne1=2 GROUP BY nom_colonne2`

Explication partitionne les résultats par champ sur lesquels il effectue les fonctions

ORDER BY

Ordonne les données selon la ou les colonnes choisies

Syntaxe : `ORDER BY nom_colonne1, nom_colonne2…`

Options `ASC` (par défaut) : ordre croissant), `DESC` : ordre décroissant

Exemple `SELECT * FROM nom_table where nom_colonne1=2 ORDER BY nom_colonne2 DESC`

Explication trie selon l'ordre de la table des caractères ASCII (chiffre, capitale, bas de casse…)

LIMIT

Limite le résultat de la requête à un certain nombre de lignes

Syntaxe : `LIMIT nombre_de_lignes`

Options `LIMIT [`*ligne_de_début*`,] `*nombre_de_lignes*

Exemple `SELECT * FROM nom_table where nom_colonne1=2 LIMIT 2, 30`

Explication Les lignes sont numérotées à partir de 0

Voici encore quelques clauses :

Tab. 4.11 : Quelques clauses

Clause	Description
`AS`	Désigne une donnée par un alias.
`LIKE`	Permet une recherche sur une partie de la valeur du champ avec des caractères génériques comme `%` ou `_`.
`RLIKE`	Permet une recherche avec des expressions rationnelles.

```
$sql="select COUNT(*) as compte, pays.nom as nation from membres,
 pays where pays.code=membres.pays GROUP BY nation";
$resultat=@mysql_query($sql,$id_link);
while($rang=@mysql_fetch_array($resultat)){
$nation=$rang['nation'];
$compte=$rang['compte'];
echo "<TR><TD><font face=\"Tahoma, Arial\" size=\"1\">$nation :
</font></td><TD>
 <font face=\"Tahoma, Arial\" size=\"1\">$compte</font></td></tr>";
}
```

Dans ce programme, vous considérez le nombre de membres de votre club par pays.

Nous utilisons la fonction `COUNT(*)` pour compter le nombre de lignes qui sont ensuite regroupées par pays, ce qui nous donne le nombre de membres par pays. Nous utilisons `AS` pour donner un alias au nombre et un autre au pays. Nous définissons le champ avec la table, ainsi `pays.code` désigne le champ *code* dans la table *pays*.

Voici une requête composée avec `LIKE` :

```
$sql="select COUNT(*) as compte from membres, pays where
membres.nom_usage LIKE 'b%'";
```

Dans cette requête, `%` remplace plusieurs caractères. Ainsi la recherche se fait sur le nombre de membres dont le nom d'usage commence par `b`. Si nous avions utilisé `'%b%'`, la requête aurait demandé à la base de trouver un nom d'usage comportant la lettre `b`. Le caractère `_`, lui, remplace un seul caractère.

```
$sql="select COUNT(*) as compte from membres, pays
 where membres.nom_usage LIKE 'b___'";
```

Dans cette requête, nous avons inséré trois blancs soulignés, cela signifie que nous comptons le nombre de membres dont le nom d'usage commence par `b` et comprend quatre lettres.

Au cours des chapitres suivants, vous retrouvez ces fonctions dans diverses applications.

4.14 Les procédures stockées

Avec sa version 5, MySQL propose l'utilisation de procédures stockées. Ce type de fonctionnalité économise du code PHP et rend les requêtes réutilisables. Cela crée une sorte d'interface entre la base de données et le script. La commande est compilée sur le serveur et est alors disponible pour des requêtes. L'application PHP appelle la procédure en lui envoyant de nouveaux paramètres, celle-ci s'exécute en une fois et renvoie un résultat. L'opération est plus rapide qu'une requête envoyée simplement par une fonction PHP car la requête n'a besoin d'être analysée qu'une seule fois. Cette version répondra à la norme SQL 2003.

4.15 SQLite

SQLite est proposée avec la version 5 de PHP. C'est une base de données sans serveur. Elle se sert de fichiers pour stocker ses données, ce qui la rend très rapide et très simple à mettre en place. Le fichier est verrouillé pour toute requête pendant un accès en écriture, ce qui rend SQLite impropre à une utilisation comportant de nombreux accès écriture.

Une des particularités de SQLite est que vous pouvez créer une base dans la mémoire pour faire une opération pendant la durée de votre script.

```
$db = sqlite_open(':memory:');
```

Imaginons que vous désirez proposer les codes postaux français sur votre site.

```
$commune=$_POST['commune'];
if ($commune){
  $db = sqlite_open('codesPostaux.db');
  // cette fonction ouvre une base existante ou en crée
  // une si elle n'existe pas
  $sql = "SELECT commune, codepostal from liste where commune LIKE
  '%$commune%'";
  $resultat = sqlite_query($db, $sql );
  echo "Résultats trouvés :<UL>"
  while ($rang = sqlite_fetch_array($resultat)){
    $commune=$rang['commune'];
    $codepostal=$rang['codepostal'];
    echo "<LI>$commune $codepostal</LI>"
  }
  echo "<UL>"
  sqlite_close($db);
```

```
}
echo <form method=\"post\" action=\"$PHP_SELF\">";
echo 'Tapez le nom de la commune<input name="commune" size="20">
<input type="submit" value="Chercher">
</form>';
```

Les droits sur votre répertoire de base de données

N'oubliez pas de donner des droits d'écriture à votre répertoire de travail.

Cette base est très facile à monter et comporte de nombreuses fonctionnalités comme la possibilité de faire des requêtes imbriquées pour des tables jointes par exemple dans le cas d'un catalogue de livre où la collection est représenté par un numéro qui est reporté dans une table collections avec le nom en clair :

```
SELECT titre FROM catalogue WHERE collection =
(SELECT code FROM collections WHERE nom = 'e-poche')
```

ou

```
SELECT catalogue.titre FROM catalogue, collections
WHERE catalogue.collection = collection.code
AND collection.nom = 'e-poche'
```

Un autre des traits de SQLite est la possibilité d'y enregistrer des fonctions PHP. Par exemple, vous pouvez décider, pour continuer l'exemple précédent de glisser une espace dans le code postal pour mettre en valeur le numéro de département comme par exemple 35 270

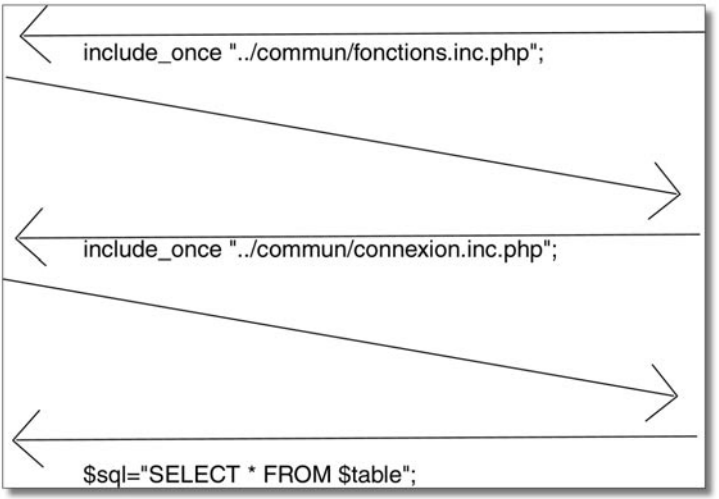
```
function formater_codepostal($codepostal) {
    return substr_replace($codepostal, ' ', 2, 0);
}
sqlite_create_function($db, 'format_code', ' formater_codepostal', 1);
$rangs = sqlite_array_query($db, 'SELECT format_code('codepostal')
from 'codePostaux.db');
```

Cette présentation succincte sert d'introduction à cette base de données qui propose avec simplicité un certain nombre de fonctionnalités très utiles tout en ayant une capacité de 2 téraoctets. Elle est adéquate pour tout site web moyen, c'est-à-dire la grande majorité des sites.

5

Le PHP et l'approche objet

Le langage PHP, rappelez-vous, est un langage procédural, c'est-à-dire que l'application parcourt le fichier PHP ligne à ligne, de droite à gauche, lisant les instructions au fur et à mesure d'une manière linéaire. PHP 4 a intégré la programmation objet d'une façon hybride, c'est-à-dire sans en intégrer toutes les implications. En PHP 4, on peut éviter d'utiliser la programmation objet et programmer ce que l'on veut. Si maintenant vous désirez exploiter les nouvelles possibilités de PHP 5, qui a particulièrement développé cet aspect, il vous faut en comprendre les multiples aspects.



▲ Figure 5.1 : *Langage procédural*

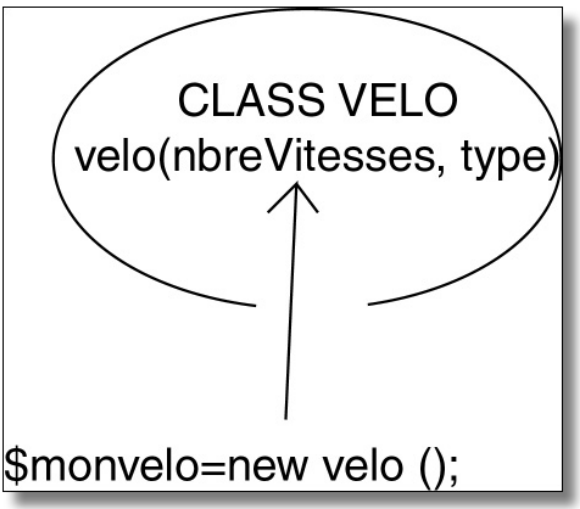
Nous verrons les différences de conception entre PHP 4 et PHP 5 en fin de chapitre pour ne pas nuire à la clarté de l'explication.

Pourquoi vouloir encombrer PHP avec une option qui lui est étrangère, telle que celle d'objet ?

5.1 Qu'est-ce que l'objet ?

Le livre que vous tenez en main est un objet ; le siège sur lequel vous êtes probablement assis est un objet. C'est un élément autonome et entier avec des propriétés et des fonctions.

C'est une capsule percée de plusieurs ouvertures. Ces capsules sont des unités indépendantes du contexte dans lequel elles sont utilisées, elles peuvent donc être utilisées dans tous les contextes. Vous pouvez voir ces capsules comme des briques de programme. Ces bouts de programmes sont donc autonomes.



▲ Figure 5.2 : *Langage à objets*

La classe est le modèle, et l'objet est une création à partir de ce modèle (une instance). La classe est le prototype et l'objet, son application dans des situations concrètes.

Cet objet a des propriétés ou des attributs. Ces propriétés sont portées par des variables à l'intérieur de la classe. Les fonctions ou méthodes de la classe concernent les actions traitées par cette classe. Une des fonctions de la civilisation est de classer différents objets.

Pour créer une classe, il faut d'abord modéliser un objet, c'est-à-dire distinguer et choisir les différentes propriétés et méthodes dont vous avez besoin pour le générer.

Prenons l'objet deux-roues ; nous allons énumérer les différentes propriétés qui le distinguent d'un autre deux-roues :

- la couleur ;
- la taille ;
- l'état.

Examinons la liste. La dernière propriété est de moins en moins pertinente, nous l'enlèverons donc de notre liste.

On peut déjà créer une classe `deux_roues`

```
class deux_roues {
   var $couleur;
   var $taille='adulte';
   var $etat="est arrêté";

   function deux_roues ($couleur, $taille, $etat){
      $this-> couleur = $couleur;
      $this-> taille = $taille;
      $this-> etat = $etat;
   }

   function affiche (){
   echo "mon deux-roues ".$this->couleur." ". $this->etat. ". Il est
   de taille " . $this->taille. " et de type " . $this->type. "
   avec ".$this->nbreVitesses. " vitesses.";
   }

   function rouletil (){
      $this->etat;
      setlocale (LC_ALL, 'fr_FR');
      $moment=strftime ("%A %e %B %Y à %H %M");
      $this->moment=$moment;
      echo "Ce deux-roues".$this->etat."$moment";
   }
}
```

Nous avons ici une classe avec les attributs et la méthode `deux_roues()`. Comme vous l'avez remarqué, cette méthode porte le même nom que la classe, ce qui la désigne comme constructeur. Cette méthode va servir à construire l'objet quand la classe sera appelée avec la méthode `new()`.

À l'intérieur de la fonction, nous employons la variable `this` accompagnée de l'opérateur `->`. La variable `this` est une variable passe-partout qui évite de donner à chaque fois un nom pour une variable qui est locale à la classe et donc temporaire. L'opérateur `->` référence une classe, une méthode ou une propriété d'une classe.

Pour appeler l'objet `deux_roues` :

```
include("deux_roues_class.php");
$mondeux_roues=new deux_roues('vert', 'enfant', 'roule');
echo "Mon deux-roues est ".$mondeux-roues ->couleur;
echo "Mon deux-roues ". $mondeux-roues ->rouletil();
```

La première ligne inclut la classe `deux_roues`. Il est conseillé de mettre une seule classe par fichier pour pouvoir s'y retrouver par la suite.

La deuxième ligne appelle la classe `deux_roues` pour en créer une instance nommée `$mondeux_roues` à laquelle j'envoie des attributs.
Dans le cas du PHP 4, le programme crée une copie de l'objet. Cette opération est consommatrice de ressources, ce qui ralentit l'opération. Le second problème est que vous agissez non pas sur l'objet lui-même mais sur sa copie.

5.2 L'héritage

Dans la catégorie, il existe plusieurs types d'objets comme vélo, moto. Ces objets sont dérivés de l'objet deux-roues. La classe deux-roues est la classe parente de la classe vélo, qui est une classe dérivée.

Avec le PHP 4, une classe ne peut hériter que d'une seule classe parente. Les héritages se font avec le mot clé ' `extends` '. La classe dérivée hérite des propriétés et des méthodes de la classe parente tout en ayant la possibilité d'ajouter les siennes.

```
class velo extends deux_roues{
var $nbreVitesses=18;
var $type="VTT";
function velo ($nbreVitesses, $type){
   $this-> nbreVitesses = $nbreVitesses;
   $this-> type = $type;
   $this-> etat = $etat;
```

```
    }
}
```

Pour appeler cette classe dérivée :

```
include("deux_roues_class.php");
include("velo_class.php");
$monvelo=new velo ();
$monvelo->etat = "roule";
$monvelo->couleur = "rouge";
$monvelo->taille = "adulte";
$monvelo->nbreVitesses = "12";
$monvelo->type = "VTC";
echo "mon vélo ".$monvelo->couleur." ".$monvelo->etat.". Il est de
taille ".$monvelo->taille." et de type ".$monvelo->type." avec
"$monvelo->nbreVitesses." vitesses.".";
```

Si la classe dérivée contient une propriété ou une méthode contenue dans la classe parente alors c'est la méthode ou la propriété cette classe dérivée qui prévaut sur celles de la classe parente.

```
class velo extends deux_roues{
   var $nbreVitesses=18;
   var $type="VTT";
function velo ($nbreVitesses, $type){
   $this-> nbreVitesses = $nbreVitesses;
   $this-> type = $type;
   $this-> etat = $etat;
   $this-> identite="mon velo ";
}
function affiche (){
   echo $this->identite.$this->couleur." ". $this->etat .". Il est de
   taille ". $this->taille ." et de type ". $this->type ." avec ".
   $this->nbreVitesses . " vitesses.";
   }
}
```

Nous utilisons ici une méthode du même nom, `affiche()`. Si nous utilisons la méthode `affiche()` de la classe `velo`, nous n'avons plus accès à la méthode `affiche()` de la classe `deux_roues`.

Maintenant, nous allons faire une classe vélomoteurs et motos.

```
class velomoteur extends deux_roues{
   var $puissance=750;
   var $type="routière";
   var $marque="Honda";
   function velomoteur ($puissance, $type, $marque){
```

```
      $this-> puissance = $puissance;
      $this-> type = $type;
      $this-> marque = $marque;
      $this-> identite="ma moto ";
   }
   function affiche (){
      echo $this->identite.$this->couleur." "
      .$this->etat.". Il est de taille "
      .$this->taille." et de type ".$this->type." avec "
      .$this->nbreVitesses." vitesses.";
   }
}
```

Cela va devenir plus difficile de s'y retrouver, et nous allons donc utiliser un ersatz de surcharge avec une sorte d'interface.

5.3 Surcharger une méthode

Surcharger une méthode ou une propriété consiste, dans la programmation objet, à donner le même nom à deux propriétés ou à deux méthodes ou plus. Cette commodité est exploitée dans des langages typés dans la mesure où ces différentes méthodes utilisent des données de type différent, ou un nombre différent de paramètres, ce qui n'est pas possible en PHP 4 mais le devient en PHP 5. Pour les propriétés de même nom, il faut qu'elles soient de type différent.

C'est pratique pour réunir dans une même classe les mêmes fonctions avec des variantes pour des situations différentes sans pour autant créer une classe dérivée.

En PHP 4, nous pouvons utiliser une astuce avec la fonction `func_num_args()`, qui donne le nombre de paramètres envoyés à la fonction. Ainsi, imaginons que les fonctions `velo()` et `velomoteur()` soient à l'intérieur de la classe `vehicule`. Dans la méthode constructeur de la classe `vehicule`, on inclut une sorte d'aiguillage conditionné par le nombre d'arguments.

```
class vehicule {
function vehicule() {
if (func_num_args()==2){
$name='velo';
$argument1=func_get_arg(0);
```

```
$argument2=func_get_arg(1);
$this->$name($argument1, $argument2);
}
elseif (func_num_args()==3){
$name='velomoteur';
$argument1=func_get_arg(0);
$argument2=func_get_arg(1);
$argument3=@func_get_arg(2);
$this->$name($argument1, $argument2, $argument3);
}
}
function velomoteur ($puissance, $type, $marque){
$this-> puissance = $puissance;
$this-> type = $type;
$this-> marque = $marque;
$this-> identite="ma moto ";
}
function velo ($nbreVitesses, $type){
$this-> nbreVitesses = $nbreVitesses;
$this-> type = $type;
$this-> etat = $etat;
$this-> identite="mon velo ";
} }
/* ensuite vous appelez la classe vehicule pour créer un objet
véhicule avec la méthode velo().*/
$monvelo=new vehicule("18", "VTT");
```

Ce type de procédé rend la classe plus simple à utiliser.

5.4 Le polymorphisme

Le polymorphisme se définit comme la possibilité pour un objet de déterminer la méthode à invoquer selon l'objet passé comme argument. Cette action est gérée au vol, en temps réel.

Dans les exemples précédents, nous avons la classe `velo` et la classe `velomoteur` ainsi que la classe parente `deux_roues` avec pour chacune la méthode `affiche()` . Le polymorphisme nous évite de choisir, le programme le fait pour nous.

```
$obj1=new velo(18, "VTT");
$obj2=new velomoteur (750, "trial", "Suzuki");

$deux_roues ->affiche($obj1);
$deux_roues ->affiche($obj2);
```

5.5 Passer un argument par référence

Avec PHP 4, l'argument passé à une fonction contient par défaut la copie de la variable, ce qui pose un problème si vous modifiez cette variable.

Référence et valeur

Quand vous créez une variable, le système alloue une case mémoire pour contenir la valeur de la variable. Quand vous passez une valeur à une fonction ou à une méthode, le programme copie la valeur de cette variable en laissant la variable originale intacte. En revanche, quand vous passez une variable par référence, la valeur qui est passée à la méthode est l'adresse de l'original ; donc, les modifications à l'intérieur de cette méthode sont faites sur le contenu de la case mémoire originale, directement sur la source.

Pour qu'un argument soit toujours passé par référence à une fonction, ajoutez une esperluette devant l'argument dans la fonction :

```
function convertir_dollar_vers_euro (&$prix) {
$prix=(float)$prix;
$prix*=0.8;
$prix .= '€';
}
$phrase= 'Cet article vaut ';
$prix='15$';
convertir_dollar_vers_euro ($prix);
$phrase.= $prix;
echo $phrase; // résultat ''Cet article vaut 12€"
```

Pour qu'un argument soit passé par défaut à une fonction qui ne le fait pas par défaut, ajoutez une esperluette devant l'argument dans l'appel à la fonction :

```
function convertir_dollar_vers_euro ($prix) {
$prix=(float)$prix;
$prix*=0.8;
$prix .= $prix.'€';
}
$phrase= 'Cet article vaut ';
$prix='15$';
convertir_dollar_vers_euro ($prix);
$phrase.= $prix;
```

```
echo $phrase; // résultat ''Cet article vaut 15$"
/*il n'a pas été converti, c'est sa copie interne à la fonction qui a
été convertie */

convertir_dollar_vers_euro (&$prix);
echo $phrase; // résultat ''Cet article vaut 12€"
```

5.6 PHP 5 et l'objet

PHP 5 rejoint Java comme langage objet en s'enrichissant des fonctionnalités d'un vrai langage objet, le rendant ainsi plus rigoureux, reproductible et plus propre.

Le pointeur

PHP 5 n'utilise plus la copie de l'argument comme PHP 4 mais se sert d'un pointeur (*handle*) du type de celui utilisé pour les fichiers. Pour continuer à utiliser les objets comme dans PHP 4, vous disposez de la méthode induite `_clone()`.

Les privilèges

Les privilèges d'accès aux méthodes et aux propriétés sont de trois types :

- `private` conditionne l'accès à la méthode ou à la propriété pour les méthodes de la classe.
- `protected` conditionne l'accès à la méthode ou à la propriété pour les méthodes de la classe ou des classes héritières ou dérivées.
- `public` ouvre l'accès à la méthode ou à la propriété pour toutes les classes.

```
private $machambre;

protected function cloturer ()
```

Le constructeur et le destructeur

Le constructeur est la méthode qui permet à la classe de construire l'objet. Dans PHP 4, elle est homonyme à la classe ; dans PHP 5, elle se nommera `_construct()` ou sera homonyme.

De la même manière, la méthode `_destruct()` permet de détruire l'objet.

D'autres fonctionnalités enrichissent cette version, qui révolutionne l'approche du développeur PHP. Nous les développons encore dans les quelques chapitres suivants, en particulier dans le chapitre intitulé **Parsez-vous XML ?**

6

L'internaute est d'abord une personne

En face de son écran, l'internaute est une personne. Aussi, le webmestre doit toucher l'internaute qui se sentira reconnu par la personnalisation et protégé par l'authentification.

La personnalisation et l'authentification entretiennent des liens étroits. Le membre du club est identifié par son nom d'usage et son mot de passe, ce qui lui donne accès à des données. Cette identification permet également de personnaliser les pages et d'y inclure le surnom de ce membre ou de donner un aspect à la page tenant compte de ses goûts. Au fur et à mesure que cet internaute se connecte, envoie des e-mails, entre des données, utilise des applications, il devient une personne pour le webmestre. Cette rencontre virtuelle peut prendre de la consistance et déboucher éventuellement sur une vraie rencontre. Il en va de même entre les membres du club. Le Web remplit alors réellement sa fonction.

Dans votre site, veillez à définir une page Contact pour vous présenter. Placez-y éventuellement votre photo. Ainsi vous ne serez pas tout à fait un inconnu pour les visiteurs. Vous direz dans cette page pourquoi vous avez créé ce site et ce que vous en attendez.

Au centre des communications et des applications existe un lieu que l'on appelle tableau de bord. Cette page permet au membre de voir les applications à sa disposition, éventuellement de recevoir des messages d'information, de modifier ses données, etc.

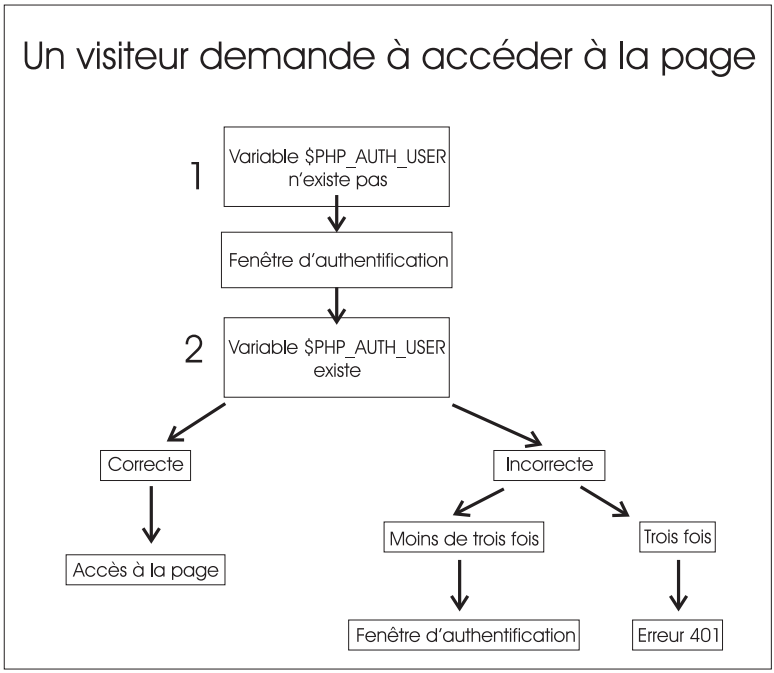
Le webmestre dispose lui aussi d'un tableau de bord pour gérer le site, ajouter des pages, regarder les statistiques ou analyser l'activité du club.

6.1 L'authentification

L'authentification en langage PHP est assez simple. Une fenêtre s'ouvre lors de l'accès à la page demandant un nom d'usage et un mot de passe. L'application vérifie ensuite l'existence des données dans la base.

Pour authentifier un visiteur, nous créerons un fichier inclus, grâce à la commande `include`, dans toutes les pages du club. Vous avez créé votre page d'accueil et un certain nombre de pages publiques mais avant d'entrer ou d'avoir envie de faire partie du club, l'internaute doit vous connaître un peu pour vous accorder sa confiance.

PHP dispose de deux variables de serveur : `$_SERVER["PHP_AUTH_USER"]` qui récupère le nom d'usage entré dans la fenêtre (login/ mot de passe) et `$_SERVER["PHP_AUTH_PW"]` qui contient le mot de passe saisi par le visiteur. Ces variables sont opérationnelles le temps de la visite du site, l'authentification ne sera donc demandée qu'une fois. Par contre, pour que le visiteur n'ait pas besoin d'entrer son identité à chaque visite du site, il vous faudra lui proposer un cookie qui enregistrera ces données sur son disque dur.



▲ Figure 6.1 : *Authentification*

Pour lancer la fenêtre d'authentification, utilisez la fonction `header()` :

```
header('WWW-Authenticate: Basic realm="Club des Bons Vivants"');
header('HTTP/1.0 401 Unauthorized');
```

La première ligne lance la fenêtre d'authentification et la deuxième aiguille le visiteur qui a cliqué sur **annuler** vers la page par défaut du serveur Apache pour une erreur 401. Cette dernière indique que l'accès à cette page n'est pas autorisé.

▲ Figure 6.2 : *Erreur 401*

La fenêtre jaillit au moment du chargement de la page. Elle offre trois possibilités à l'internaute :

1. entrer le bon mot de passe et le bon login ;

2. entrer un mauvais mot de passe ;

3. cliquer sur **annuler**.

Si nous laissions les lignes en l'état, le visiteur reviendrait constamment sur cette fenêtre d'authentification, quels que soient les nom d'usage et mot de passe saisis. S'il cliquait sur le bouton **annuler**, il arriverait sur la page d'erreur 401.

Il nous faut donc comparer les informations aux données de la table *membres* et statuer sur la réponse. Si elle est vraie, l'internaute ne verra

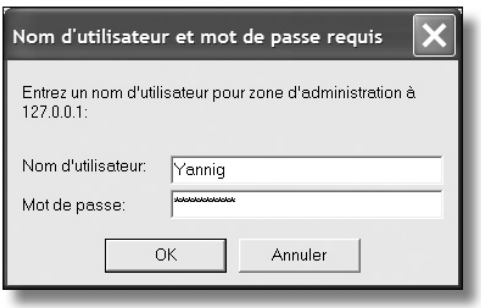
plus apparaître cette fenêtre. Si elle est fausse, il verra la fenêtre réapparaître encore deux fois.

Pour les cas 2 et 3, nous renverrons l'internaute vers le formulaire d'inscription au club.

Le processus

Le programme d'authentification est chargé à chaque requête sur une page du club. L'interrogation porte sur l'existence des variables `$PHP_AUTH_USER` et `$PHP_AUTH_PW`. Si ces variables n'ont pas été initialisées, c'est que la fenêtre d'authentification ne s'est encore jamais présentée à l'internaute. Si elles existent, le programme vérifie dans la base de données que les valeurs de ces deux variables sont présentes dans la table *membres*.

Quand l'internaute tape le nom d'usage et le mot de passe, le programme est rechargé avec ces deux variables puis entre dans le bloc conditionnel adéquat. Un bloc s'exécute en cas d'existence des variables, l'autre dans le cas où elles n'existent pas. Sur le schéma, le visiteur commence son cheminement par l'étape 1 s'il n'a jamais été authentifié, par l'étape 3 s'il a déjà été identifié.



◄ Figure 6.3 : *Cheminement de l'authentification*

Dans le schéma, nous remplacerons l'erreur 401 par une redirection vers la page d'inscription. Ensuite, nous introduirons la variable `$fois` en session, ce qui limitera le nombre de fois où l'internaute pourra taper son mot de passe. La limitation du nombre de tentatives est introduite pour des raisons de sécurité. Cela protège contre les logiciels de type Force

Brute qui envoient en rafales un grand nombre de noms d'usage et de mots de passe jusqu'à trouver l'accès, comme un serrurier le fait avec des passes.

Les mots de passe

Combien d'internautes utilisent des mots de passe faciles à mémoriser et à deviner comme toto ou leur date de naissance ? Davantage qu'on ne le pense. Ils espèrent ainsi les mémoriser plus vite.

Si l'internaute préfère ne pas entrer de mot de passe et cliquer sur **annuler**, ce peut être pour deux raisons :

- Il ne le connaît pas ou l'a oublié.
- Il n'est pas membre.

La première authentification

```
<?
if (!isset($_SERVER['PHP_AUTH_USER'])) {
header('WWW-Authenticate: Basic realm="Club des Bons Vivants"');
header('HTTP/1.0 401 Unauthorized');
$auth=3;
include "inscription_p1.php";
exit;
}
?>
```

La première ligne inclut le fichier de connexion à la base de données. Les deux lignes suivantes envoient l'en-tête HTTP d'authentification.

Les trois lignes qui suivent seront exécutées si le visiteur clique sur **annuler** dans la fenêtre d'authentification. Elles seront exécutées aussi en cas de trois tentatives infructueuses.

Nous reprenons la variable `$auth` que nous avons utilisée dans le chapitre précédent pour afficher une annonce dans le formulaire d'inscription. Nous lui donnons la valeur 3. Ces différentes valeurs nous permettent de différencier les informations à afficher dans ce formulaire. Ainsi nous

avions inséré ce code dans le fichier *inscription_p1.php* à la suite des expressions pour la valeur 1 et la valeur 2.

Reportez-vous à la partie *Ouvrir une fenêtre avec une fonction Javascript* du chapitre précédent *MySQL en action*.

La racine du site

Indiquez à chaque fois l'adresse intégrale (ou absolue) de votre fichier inclus car la place du fichier appelant peut varier et invalider l'adresse relative du fichier appelé. Pour cela, il suffit de créer un fichier de fonction qui sera appelé par toutes les pages, nommé *fonctions.inc.php*, et inséré dans le répertoire *commun*. Dans ce fichier, vous inclurez toutes les fonctions et la constante de la racine WWW du site que vous nommez RACINEWWW, et la racine du chemin local que vous nommez RACINE. Attention, les constantes sont pratiques en début de programme puisque vous changez facilement leur valeur si vous déménagez votre site mais elles ne sont pas extrapolées, donc vous ne pouvez les utiliser à l'intérieur de guillemets.

```
define ("RACINEWWW",
"http://www.happyneurone.com/happyneuron");
define ("RACINE", "home/happy/htdocs/happyneuron");
```

Il vous suffira ensuite de mettre `include RACINEWWW."commun/connexion.inc.php";` pour inclure le fichier de connexion. Pour avoir le chemin local des fichiers, créez une erreur de syntaxe et notez le message d'erreur, il indique le chemin local jusqu'au fichier en question. Autrement, le plus simple est d'utiliser la variable définie `$DOCUMENT_ROOT`.

```
define ("RACINE", $_SERVER["DOCUMENT_ROOT"]);
```

Maintenant que nous avons étudié le cas où le visiteur n'a pas encore été sollicité pour insérer son nom d'usage et son mot de passe, voyons celui où les informations sont entrées et où il faut les vérifier.

La vérification de l'identité de l'internaute

L'internaute a entré un nom d'usage et un mot de passe, pour cette page ou une autre page du site. Le résultat est analysé en regard des données de la base. La réponse est correcte ou incorrecte. Si elle est correcte, l'accès est libre à la page, si elle est incorrecte, nous répétons le processus trois fois puis présentons la page d'inscription à l'internaute d'où il peut s'inscrire au club ou disposer d'un lien vers la page *oubli.php* pour qu'il réponde à la question secrète :

```
elseif (isset($_SERVER["PHP_AUTH_USER"])) {
$sql = "SELECT * FROM membres WHERE nom_usage=$_SERVER['PHP_AUTH_USER']
 and mot_passe=$_SERVER['PHP_AUTH_PW']";
$resultat=mysql_query($sql,$id_link);
$nombre = @mysql_num_rows($resultat);
if (!$nombre || $nombre<1) {
header('WWW-Authenticate: Basic realm="Club des Bons Vivants"');
header('HTTP/1.0 401 Unauthorized');
$auth=3;
include "http://127.0.0.1/BonsVivants/inscription_p1.php";
exit;
}
}
```

Nous allons insérer une variable de session nommée `fois`. Nous l'initialiserons à 1 dans le fichier *fonctions.inc.php*. Maintenant, remplaçons `include` par `include_once` puisque les valeurs des variables à l'intérieur de ces fichiers *.inc.php* sont immuables. Ainsi la variable `$fois`, que nous pourrons utiliser d'autres fois, ne sera initialisée qu'une fois. Ensuite, si le visiteur entre un mot de passe et qu'il n'est pas correct, la demande d'authentification lui est renvoyée et la variable `fois` est augmentée de 1. À la troisième mauvaise réponse, le visiteur est aiguillé vers la page d'inscription au club.

```
include_once "../commun/fonctions.inc.php";
include_once RACINE."/commun/connexion.inc.php";
session_register("fois");
if (!isset($_SERVER["PHP_AUTH_USER"])) {
   header('WWW-Authenticate: Basic realm="Club des Bons
          Vivants"');
   header('HTTP/1.0 401 Unauthorized');
   $auth=1;
   include "inscription_p1.php";
   exit;}
```

```
elseif (isset($_SERVER["PHP_AUTH_USER"])) {
   $fois++;
  $sql = "SELECT * FROM membres WHERE
 nom_usage=$_SERVER['PHP_AUTH_USER']
  and mot_passe=$_SERVER['PHP_AUTH_PW']";
  $resultat=mysql_query($sql,$id_link);
   $nombre = @mysql_num_rows($resultat);
   if (!$nombre || $nombre<1) {
      if ($fois==3){
         $auth=3;
         include "inscription_p1.php";
         exit;
      }
      header('WWW-Authenticate: Basic realm="Club des Bons
             Vivants"');
      header('HTTP/1.0 401 Unauthorized');
      $auth=3;
      include "inscription_p1.php";
      exit;
   }
}
```

`include_once` vérifie si le fichier a déjà été inclus. Si c'est le cas, elle ne le rappellera pas une deuxième fois.

Le cookie

Vous pourriez vous contenter de ce résultat mais nous ajouterons deux améliorations. La première est l'utilisation d'un cookie, la seconde, le stockage de données dans la session.

Nous verrons un peu plus loin comment créer le cookie. Ici, pour éviter au visiteur de passer par la fenêtre d'authentification à chaque connexion au club, nous récupérerons ses données sur un petit fichier nommé *cookie*, dans son disque dur. Cela se fera uniquement s'il l'a demandé, comme nous le verrons plus tard.

Dans le fichier d'authentification, il suffit d'ajouter un petit bloc qui récupère nom d'usage et mot de passe pour court-circuiter la fenêtre. Dans le cookie, nous insérons les valeurs `$alias` et `$sauf_conduit` qui contiennent les valeurs `$nom_usage` et `$mot_passe`. Nous n'utilisons pas les mêmes variables pour les différencier. Comme une variable contenue dans un cookie est instantanément disponible dans le langage PHP, si la variable `$alias` est disponible, c'est que l'internaute a accepté un cookie.

Listing 6.1 : *Authentification : authentification.inc.php*

```
if(isset($_COOKIE["alias"])){
  $alias=$_COOKIE["alias"];
  $sauf_conduit =$_COOKIE["sauf_conduit"];
  $sql = "SELECT * FROM membres WHERE nom_usage='$surnom'
    and motdepasse='$sauf_conduit'";
  $resultat=mysql_query($sql,$id_link);
  $nombre = @mysql_num_rows($resultat);
  if ($nombre==1) {
   $_SERVER["PHP_AUTH_USER"]=$alias;
   $_SERVER["PHP_AUTH_PW"]=$sauf_conduit;
  }
  else {
    setcookie(sauf_conduit,$sauf_conduit,time()-30);
    setcookie(alias,$alias,time()-30);
    header('WWW-Authenticate: Basic realm="Club des Bons Vivants"');
   header('HTTP/1.0 401 Unauthorized');
    $auth=3;
    include "inscription_p1.php";
    exit;
  }
}
```

Nous interrogeons les valeurs des variables contenues dans le cookie afin de les confronter aux valeurs des colonnes dans la base de données. Si les valeurs sont exactes, nous les insérons dans les variables `$_SERVER["PHP_AUTH_USER"]` et `$_SERVER["PHP_AUTH_PW"]`. Ainsi elles seront testées à nouveau de manière transparente pour l'utilisateur et la page sera chargée. Si la réponse est inexacte, nous effaçons le cookie. Pour effacer un cookie, nous en créons un nouveau avec une valeur de temps située dans le passé que nous insérons dans le champ expiration. *time* donne le nombre de secondes du 1er janvier 1970 jusqu'au moment présent. En y retirant 30 secondes, nous donnons une valeur de temps du passé. Ces deux nouveaux cookies vont écraser les anciens et s'effacer à leur tour en raison de leur date d'expiration.

Pour que le membre puisse choisir l'authentification automatique par cookie, nous créerons une page permettant de recevoir un cookie ou d'enlever le cookie existant. Nous aurions pu mettre d'office un cookie mais il faut savoir que l'internaute ne se connecte pas seulement depuis son ordinateur personnel. Il peut le faire depuis son lieu de travail ou d'un cybercafé. Dans ce cas, il n'est pas souhaitable de stocker un cookie sur le disque dur de la machine. D'autre part, l'internaute peut refuser les cookies grâce aux options de son navigateur.

L'usine à biscuits

Le biscuit (cookie) peut être fabriqué par un fichier qui s'affiche dans une petite fenêtre. Le lien vers cette page sera inséré dans la page d'inscription et le tableau de bord du membre.

Dans le tableau de bord, nous avons déjà inséré la fonction Javascript `ouvrir_vasistas()`. Il suffit ici de la réutiliser en changeant son nom et les paramètres.

Dans le fichier *tableaudebord.php*, insérez :

```
 echo ' <a href="#" onClick="ouvrir_lucarne
('fabrique_biscuits.php?biscuit=oui','vasistas','menubar=no,
scrollbars=no,statusbar=no, width=300,height=200')">
Pour insérer un cookie</A>.';
```

Le signe # est un signe d'ancrage dans une page. Si vous souhaitez par exemple accéder à un paragraphe à l'intérieur de la page, vous ajoutez # avec le lien. Vous ne pouvez laisser un lien sans caractère. Pour rester sur la même page, en Javascript, nous utilisons une astuce : cet ancrage qui ne va nulle part permet de garder la page ouverte bien que nous cliquions sur le lien vers une autre page. Ensuite, vous créez le fichier *fabrique_biscuits.php*.

En haut du fichier avant les en-têtes HTML, créez le cookie :

```
<?
if (biscuit='oui'){
  setcookie(alias,$alias,time()+ (60*60*24*365*20));
////Il va en prendre pour 20 ans!
  setcookie(sauf_conduit,$sauf_conduit, time()+(60*60*24*365*20));
}
  session_start();
?>
```

Puis, dans le corps de la page, créez un bloc dont l'exécution est liée au fait qu'il s'agit de la première entrée de l'internaute dans la page. Le bloc est testé avec la non existence de la variable `$biscuit`. Dans cette partie, vous expliquez l'intérêt du cookie et les réglages nécessaires dans le navigateur internet :

```
if (!biscuit){
echo "Vous n'aimez pas entrer vos nom d'usage et mot de passe à
chaque accès au club, pourquoi ne pas utiliser un cookie ? Un
cookie est un petit fichier qui garde les paramètres de votre
identité sur votre disque dur.
Comment faire ? <LI> <b>Internet Explorer</b> : allez dans le menu
outils et cliquez sur option internet. Ensuite vous cliquez sur
l'onglet \"sécurité\". Dans la fenêtre, activez l'image du globe
(internet) et cliquez sur \"personnaliser le niveau\". Ensuite
vous cochez les cases dans la catégorie \"cookies\" puis sur
\"autoriser les cookies par session (non stockés)\" et \"autoriser
les cookies stockés sur votre ordinateur\".
<LI> <b>Netscape</b> : dans le menu \"édition\", vous cliquez sur
\"préférences\". Ensuite vous sélectionnez \"avancées\" et vous
cochez \"Accepter tous les cookies\".<p>";
echo " <a href=\"$_SERVER['PHP_SELF']?biscuit=oui\">
Pour enregister le cookie</A>.";
}
```

Sinon, l'internaute a déjà cliqué pour enregistrer le cookie :

Listing 6.2 : *Ajouter un cookie : fabrique_biscuits.php*

```
else {
echo "Le cookie est créé, désormais vous n'aurez plus besoin de
 vous identifier dans le club des Bons Vivants. Cependant, si vous
 voulez nettoyer de temps en temps votre boîte à biscuits,
 il vous suffira de revenir sur cette page. N'oubliez pas de <A
 HREF=\"#\"onClick=\"window.close()\"> fermer la porte</A>
 en partant.";
}
?>
```

setcookie()

Cette fonction permet de créer un cookie

Syntaxe : `setcookie (string nom[, string valeur, int expiration, string chemin, string domaine, int sécurisé])`

Explication la valeur expiration est un timestamp Unix (nombre de secondes depuis le 1/1/70), sécurisé est possible (1) pour une adresse https://

La conservation des données personnelles

Profitons que le membre du club s'est identifié pour conserver quelques variables indispensables au cours de sa visite. Les trois données intéressantes sont la valeur de la colonne clé, son nom d'usage et son e-mail. La clé est une colonne indexée , elle constitue donc un bon outil pour sélectionner, modifier ou supprimer les données du membre. Le nom d'usage du club permet de personnaliser les pages en s'adressant directement à lui. L'adresse e-mail est indispensable pour envoyer un courriel de confirmation suite à certaines actions effectuées par cette personne. Nous plaçons cette extraction de données après la vérification du nom d'usage et du mot de passe :

```
$sql = "SELECT * FROM membres WHERE nom_usage=$_SERVER['PHP_AUTH_USER']
 and mot_passe=$_SERVER['PHP_AUTH_PW']";
$resultat=mysql_query($sql,$id_link);
$rang=mysql_fetch_array($resultat);
$nom_usage=$_SERVER["PHP_AUTH_USER"];
session_register("nom_usage ");
if (!$clef){
$clef=$rang['clef'];
session_register("clef");
}
if (!$email){
$email=$rang['email'];
session_register("email");
}
```

Afin de conserver les données, vous pouvez les enregistrer dans la session courante. Pour cela, il suffit d'utiliser la fonction `session_register()`. Attention, vous y mettez le nom de la variable à enregistrer et non la variable elle-même. N'oubliez pas les guillemets.

Session_start()

Pour récupérer les variables enregistrées dans la session courante, il faut ajouter à chaque début de fichier la fonction `session_start()`. Cette fonction crée une session ou réveille la session courante.

L'enregistrement d'une variable dans une session constitue la façon la plus simple et la plus sûre de conserver des valeurs en passant d'une page à une autre. Il en existe d'autres.

Les formulaires permettent d'envoyer des données sur une autre page. Le bouton peut être une image et le formulaire contenir des champs qui ne seront pas affichés, les champs cachés (`hidden`). Ces champs sont dissimulés mais l'internaute peut les voir en affichant la source de la page. Les informations transportées par ce moyen ne doivent pas être confidentielles ou stratégiques car une personne mal intentionnée peut travestir ces champs et envoyer des données malveillantes.

Un lien peut contenir des variables. Ces variables sont placées après un point d'interrogation à la fin de l'URL. Elles sont disposées de part et d'autre d'un signe égal, à gauche le nom de la variable, à droite sa valeur. Les variables sont séparées par un signe `&`. Les données dans l'URL doivent être codées avec la fonction `urlencode()`.

Voici un exemple d'URL pour comptabiliser le nombre de liens vers une page :

```
http://www.bonvivants.com/suivi.php?page=http://www.agirici.org
```

Dans le fichier *suivi.php*, vous ajoutez un compteur que nous verrons dans le chapitre suivant et aiguillez l'internaute sur le site cible en tapant cette ligne :

```
header("Location: $page");
```

Vous pouvez aussi le faire avec un cookie, comme nous venons de le voir. Cependant, la session reste le moyen le plus simple et le plus transparent pour l'utilisateur.

6.2 Le tableau de bord du membre

Le tableau est une page *index.php* dans le répertoire *club* qui liste les actions possibles et les renseignements concernant le membre. Sur cette page vont s'afficher des liens pour que le membre puisse modifier son profil ou le supprimer. En outre, cette page constitue un carrefour pour toutes les applications et communications à l'intérieur du club. Dans cette étape, nous n'avons pas encore d'applications mais nous allons voir comment le membre du club peut modifier ou supprimer son profil.

Modifier ou supprimer un profil

Modifier ou supprimer un profil est simple. Cette opération s'effectue, côté programmeur, en utilisant les commandes SQL `DELETE` et `UPDATE`, respectivement pour la suppression et la mise à jour :

```
$sql="DELETE FROM membres WHERE clef='$clef'";
mysql_query($sql, $id_link);
$sql="UPDATE membres SET email='$email'";
mysql_query($sql, $id_link);
```

Il s'agit maintenant de mettre ces commandes en œuvre. Pour une mise à jour, nous avons étudié le formulaire d'inscription et, à l'occasion de la vérification d'un menu à sélection multiple, complété le formulaire avec les valeurs par défaut des différents champs. Il nous suffit ici de créer un lien vers ce formulaire en envoyant une variable par le lien. Le formulaire s'affichera ainsi avec les valeurs des données personnelles du membre. Ce dernier pourra alors les modifier.

Nous effectuerons également un ajout dans le traitement du formulaire pour introduire la requête SQL `UPDATE` au lieu de `INSERT`. Il faudra prévoir de rassurer le membre en lui confirmant la mise à jour de ses données par un message de remerciement.

Vous objecterez peut-être qu'il manque les champs *nom d'usage*, *mot de passe*, *question secrète* et *réponse*. Le nom d'usage n'est pas modifiable mais pour les autres champs, il suffit d'ajouter un bloc conditionnel dans le formulaire pour les faire apparaître en même temps. À la fin du formulaire, nous insérons ou non un lien vers une adresse e-mail pour demander le changement de nom d'usage par le webmestre.

Changement du nom d'usage

Le changement de nom d'usage n'est pas souhaitable car c'est par le nom d'usage que les membres se connaissent entre eux. C'est aussi votre manière de connaître ce membre et de suivre l'évolution de son profil, comme nous le verrons plus loin dans ce chapitre. Vous pouvez accepter de modifier une lettre dans le nom d'usage ou de le transformer en un nom approchant. Il sera alors

nécessaire de l'annoncer à tous les membres par un courriel. Nous ajouterons cette application dans le tableau de bord du webmestre.

Première étape

Cette étape consiste à créer les liens dans le tableau de bord :

```
<font size="3">Mon profil</font>
<ul>
 <li><a href="inscription_p1.php?action=modifier">
modifier mon profil</a>
 <li><a href="demission.php">démissioner du Club</a>
</ul>
```

Dans le lien vers le fichier *demission.php*, introduisons une fonction Javascript pour que le membre confirme son désir de démissionner avant de le supprimer de la base :

```
<li><a href="demission.php" onClick="return confirmDem()">
démissioner du Club</a>
```

La fonction Javascript s'écrit ainsi :

Listing 6.3 : *le tableau de bord club/index.php*

```
<script LANGUAGE="JavaScript">
<!--
function confirmDem()
{
var accord=confirm("Vous voulez
 vraiment démissionner et perdre votre statut de membre du Club ?");
if (accord)
    return true ;
else
    return false ;
}
// -->
</script>
```

Si le membre clique sur **annuler**, l'action ne sera pas effectuée. Si le bouton OK est cliqué, le chargement de la page de démission suivra.

Deuxième étape

Commençons par le plus facile, créer le fichier *demission.php*.

Nous allons auparavant copier dans le fichier *fonctions.inc.php* les différentes fonctions que nous avons créées dans les chapitres précédents. Ainsi, copiez-y le contenu du fichier *majuscules.inc.php*.

Ensuite, dans le fichier *demission.php*, commencez par inclure les fichiers *.inc* :

```
include_once "../commun/fonctions.inc.php";
include_once RACINE."/commun/connexion.inc.php";
```

Nous tricherons temporairement en conservant une trace du membre. Nous copions son prénom et son adresse e-mail dans une table. Cette action nous permettra de lui envoyer un courriel afin de faire, avec tact et courtoisie, une petite enquête sur les raisons qui l'ont poussé à démissionner. Cette enquête nous aidera à améliorer le site. Le membre démissionnaire sera heureux d'être reconnu et de pouvoir donner son avis. Une fois que vous aurez envoyé ce courriel, détruisez les données de cet internaute. Sur le Web, le respect de la volonté du membre du club est essentiel.

Vous allez créer une table supplémentaire appelée *demission* avec les colonnes `clef`, `prenom`, `email` et `date`. Le champ *date* est une colonne de type `TIMESTAMP(8)`, elle sera donc remplie automatiquement par la date courante. Nous avons stocké la variable `email` dans le fichier d'authentification :

```
$sql="select prenom from membres where clef='$clef'";
$resultat=mysql_query($sql, $id_link);
$rang=mysql_fetch_array($resultat);
$prenom=$rang['prenom'];
$sql="delete from membres where clef='$clef'";
mysql_query($sql, $id_link);
majuscules($prenom);
echo "$prenom, vous ne faites plus partie du club des Bons
 Vivants mais quelle que soit la raison de votre démission,
 vous y serez toujours le bienvenu. Nous vous remercions
 pour la confiance que vous nous avez témoignée. Nous nous
 permettrons de vous envoyer un email avec une enquête de
 satisfaction auquel vous pourrez répondre pour nous permettre
 de nous ajuster à vos envies et à vos besoins.";
```

Avec ce message, l'ex-membre attendra votre courriel. S'il est mécontent des services du club, vous pourrez l'écouter et tenter de résoudre les problèmes. Chacun, sur Internet, a besoin de se sentir unique.

Définissons maintenant le formulaire d'inscription. Il est presque adapté mais nous devons ajouter des champs à la suite des autres champs, juste avant le bouton de soumission du formulaire :

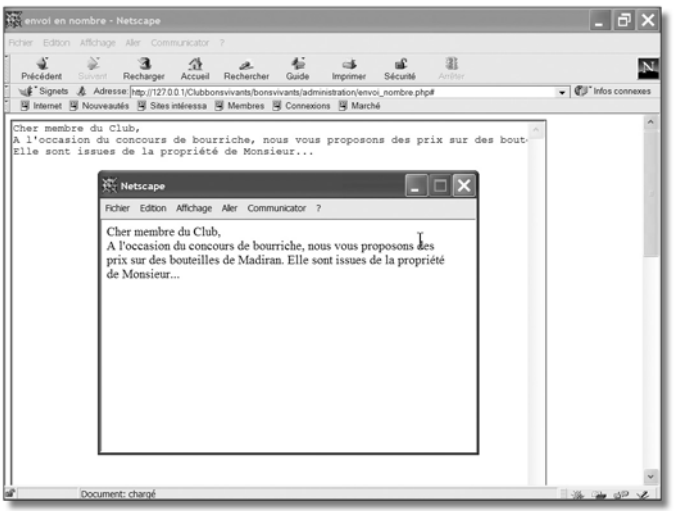
```
if ($_GET["action"]=='modifier'){
$action=$_GET["action"] ;
echo "<td>Votre nouveau mot de passe</td>";
echo "<td><input type=\"password\" name=\"mot_passe\"
 value=\"$mot_passe\"></td>";
echo "</tr>";
echo "<tr>";
echo "<td>Retapez votre mot de passe</td>";
echo "<td><input type=\"password\" name=\"mot_passebis\"
 value=\"$mot_passe\"></td>";
echo "</tr>";
echo "<tr>";
echo "<td>Votre question secrète en cas d'oubli</td>";
echo "<td><input type=\"text\" name=\"question\"
 value=\"$question\" size=\"50\"></td>";
echo "</tr>";
echo "<tr>";
echo "<td>Votre réponse</td>";
echo "<td><input type=\"text\" name=\"reponse\"
 value=\"$reponse\"></td>";
echo "</tr>";
}
```

Cela ne suffira pas car il nous faut ajouter la fonction `session_start()` en haut de la page et récupérer les valeurs des champs avec une requête SQL :

Listing 6.4 : *le formulaire pour modifier*

```
session_start();
if (($_GET["action"]=='modifier'){
$action=$_GET["action"] ;
$sql="select * from membres where clef='$clef'";
$resultat=mysql_query($sql, $id_link);
$rang=mysql_fetch_array($resultat);
$clef=$rang['clef'];
$nom_usage=$rang['nom_usage'];
$mot_passe=$rang['mot_passe'];
$prenom=$rang['prenom'];
$code_postal=$rang['code_postal'];
$ville=$rang['ville'];
```

```
$pays=$rang['pays'];
$email=$rang['email'];
$question=$rang['question'];
$reponse=$rang['reponse'];
}
```



▲ Figure 6.4 : *Le formulaire renseigné*

Pour finir cette étape, ajoutez le code Javascript de la deuxième page du formulaire sur la première page, afin de vous assurer que les champs sont bien remplis. Pour cela, il faut créer un bloc conditionnel `if ($action=='modifier')` qui permet d'afficher le Javascript avec `echo` et des guillemets simples, en prenant garde de mettre des barres obliques inversées pour échapper les apostrophes dans le code Javascript.

Troisième étape

Nous allons modifier ou créer le fichier de traitement des données. Modifier la deuxième page serait fastidieux, il vaut mieux ici créer un nouveau fichier. Ce choix oblige à ajouter un aiguillage dans la première page du formulaire pour que le paramètre `action` dans la balise `form` puisse envoyer les données dans ce nouveau fichier que nous appellerons *traitement_modif_profil.php*. Un nom long s'impose ici car il vaut mieux

être explicite, sinon vous vous perdrez dans vos noms de fichiers, dans six mois, quand vous aurez oublié tout cela. Imaginez aussi qu'un collègue reprenne votre travail. Il faut qu'il comprenne les correspondances entre les fichiers et les actions. En haut du fichier, insérez :

```
session_start();
$clef=$_SESSION['clef'] ;
$nom_usage=$_POST['nom_usage'] ;
$prenom =$_POST['prenom'] ;
$code_postal=$_POST['code_postal'] ;
$ville=$_POST['ville'] ;
$pays=$_POST['pays'] ;
$email =$_POST['email'] ;
$question =$_POST['question'] ;
$reponse=$_POST['reponse'] ;
$sql="UPDATE membres SET nom_usage='$nom_usage',
 mot_passe='$mot_passe', prenom='$prenom',
 code_postal='$code_postal', ville='$ville', pays='$pays',
 email='$email', question='$question', reponse='$reponse'
 WHERE clef='$clef'";
$resultat=mysql_query($sql, $id_link);
```

Vérifiez que la mise à jour a été effectuée en regardant le résultat retourné. Si `$resultat` retourne `FALSE`, la mise à jour a échoué. Dans ce cas, envoyez-vous un message aussi précis que possible et avertissez le membre en lui assurant que l'erreur sera bientôt réparée. En cas de succès, il vous suffit d'afficher dans la page suivante un message informant que la mise à jour a été effectuée.

6.3 Le tableau de bord d'administration du site

L'application PHPMyAdmin simplifie beaucoup la gestion de votre base de données, cependant il est recommandé d'avoir un tableau de bord contenant les principales requêtes SQL, les statistiques de vos pages et le suivi de votre site.

Il fonctionne sur le même principe que le tableau de bord du membre. Il est situé dans un autre répertoire que vous nommerez *administration*, dont l'accès est restreint à vous-même. Sur un système Unix, vous pouvez restreindre l'accès à un répertoire par deux fichiers qui travaillent ensemble : *.htaccess* et *.htpasswd*. Ces fichiers qui comportent un point

avec rien devant sont invisibles pour les navigateurs, le second est crypté et situé au-dessus du répertoire *www* ou *htdocs*.

L'accès restreint d'un répertoire avec .htaccess

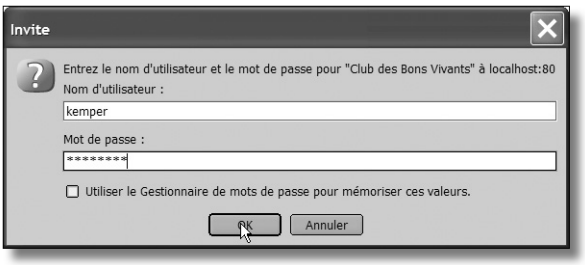
Placez le premier fichier que vous nommez *.htpasswd* au-dessus de votre répertoire web appelé *www* ou *htdocs*, selon la configuration du serveur. Dans ce fichier, écrivez `lelogin:lemotdepasse`. Attention, le mot de passe est crypté. En fin d'ouvrage, nous vous donnons un site qui crypte votre mot de passe.

Si vous souhaitez protéger un deuxième répertoire, faites la même chose en ajoutant les nouveaux `login:motdepasse` sur la ligne du dessous…

Le deuxième fichier que vous nommez *.htaccess* est conçu ainsi :

```
AuthUserFile /chemin/complet/.htpasswd
AuthGroupFile /dev/null
AuthName "zone d'administration"
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>
```

En lieu et place de `/chemin/complet`, mettez le chemin qui conduit à ce fichier. Si vous ne le connaissez pas, générez une erreur sur une application en langage PHP dans le répertoire web, elle affichera le chemin complet. Il vous suffit ensuite de le raccourcir de ce qui suit le répertoire web.



▲ Figure 6.5 : *.htaccess*

Enregistrez ensuite les fichiers et envoyez-les sur le site en mode texte.

Cette recette restera valable pour protéger le répertoire de PHPMyAdmin. Quiconque accédera à ce répertoire se verra demander un nom d'usage et un mot de passe.

Assurez-vous que votre hébergeur accepte les fichiers *.htaccess* et *.htpasswd* car certains ne le proposent pas. Vous remarquerez que si vous cryptez plusieurs fois le même mot, le résultat ne sera jamais le même. Ne vous inquiétez pas, c'est une particularité de ce mode de cryptage.

Le tableau de bord

Pour l'instant, nous allons créer le tableau de bord en y mettant deux applications. Dans les chapitres suivants, nous en ajouterons d'autres.

Les internautes sont volatiles. Vous vous promenez sur Internet et voyez un site qui vous plaît. Il vous propose une inscription gratuite à un club. Ça ne vous engage à rien alors pourquoi ne pas vous inscrire. Le lendemain, vous aurez oublié cette inscription. En changeant de fournisseur d'accès, vous oublierez bien sûr de modifier votre adresse e-mail dans ce club.

Pour éviter cela, il faut envoyer une lettre mensuelle à vos adhérents avec en bas du message, un lien pour la désinscription. Cet envoi régulier vous permettra de rappeler votre club à leur bon souvenir et de vérifier que l'adresse e-mail est toujours valide.

Commençons par créer un outil d'envoi en nombre puis un autre pour enlever de la table *membres* les membres aux coordonnées obsolètes.

L'envoi en nombre

Il s'agit d'une boucle sur le résultat d'une requête avec l'emploi de la fonction `mail()` :

```
<?
include_once "../commun/connexion.inc.php";
include_once "../commun/fonctions.inc.php";
if ($message){
   $sql="select * from membres";
   $resultat=mysql_query($sql,$id_link);
while($rang=mysql_fetch_array($resultat)){
   $nom=$rang['nom_usage'];
```

```
   $nom=majuscules($nom);
   $motdepasse=$rang['motdepasse'];
   $destinataire=$rang['email'];
   $message.="\n\nVotre mot de passe pour le club est $motdepasse.";
   $entetes = "From: LE CLUB DES BONS VIVANTS
    <contact@bonsvivants.com>\n";
   mail ("$destinataire","LA LETTRE DES BONS VIVANTS",$message,
    $entetes);
}
}
?>
```

Le courriel n'est pas formaté en *HTML*, pour que tous les destinataires puissent le lire, quel que soit leur équipement. Ensuite, vous créez le formulaire pour envoyer le message à ce programme avec un champ *textarea* :

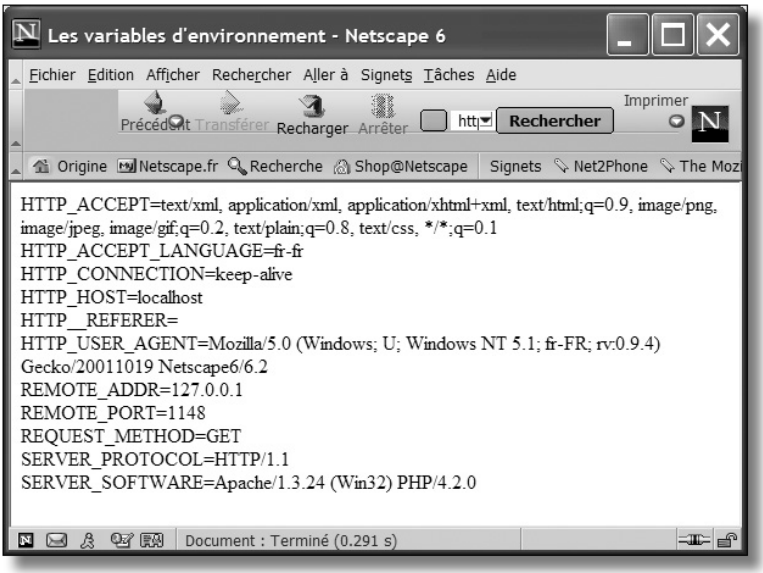
```
<?
if (!$_POST["message"]){
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
echo "<textarea cols=80 rows=80 name=\"message\"></textarea>";
echo "<input type=\"submit\" value=\"Envoyer\">";
echo "</form>";
echo "<A HREF=\"#\" onClick=\"ouvrir_message()\">Visualiser
 le message</a>";
}
else {
echo "Le message a été envoyé aux membres du club";
}
?>
```

Nous allons créer la fonction Javascript qui ouvre la fenêtre de visualisation du message. Cette fonction a été créée parce qu'il est difficile de repérer les fautes de frappe ou d'orthographe dans un champ de formulaire :

Listing 6.5 : *Formulaire pour l'envoi en nombre envoi_nombre.php*

```
<title>envoi en nombre</title>
<SCRIPT LANGUAGE="JavaScript">
<!--
function ouvrir_message() {
var hauteur=screen.height;
hauteur*=80/100;
var largeur=screen.width;
largeur*=80/100;
FenetreAfficher= window.open('','lecture','menubar=yes,
```

```
scrollbars=yes,statusbar=no, width=' + largeur + ', height='
 + hauteur +',top=0,left=0');
message= document.forms[0].message.value;
message_secret=message;
message = message.replace(/\n/,"<BR>\n");
message +="<P><A HREF=\"Je ferme la fenêtre\"
 onClick=\"window.close()\">Je ferme la fenêtre</a>\n";
FenetreAfficher.document.write(message);
}
-->
</script>
```



▲ Figure 6.6 : *Fenêtre de visualisation*

Dans ce code, nous utilisons les propriétés de l'objet `screen` : `height` et `width` (hauteur et largeur) à 80 %. La fenêtre n'occupera qu'une partie de l'écran, quelle que soit la définition de l'écran de l'internaute. Un peu plus loin, nous ajoutons une balise <BR> aux fins de ligne. Enfin, nous affichons un lien pour fermer la fenêtre.

Le cryptage simple de la clé

Nous ajouterons un lien dans le courriel pour que les personnes puissent se désinscrire simplement du club. Cette démission se fera en deux temps. Le lien dans l'e-mail enverra la clé de la table cryptée. La fonction que nous vous donnons vaut pour quelqu'un qui n'a pas de module de cryptage sur son serveur. C'est une clé facile à décrypter pour qui veut y passer un peu de temps mais qui sera suffisante pour un site personnel. Il est possible également d'envoyer la clé avec le mot de passe dans le lien du tableau de bord et dans le lien sur l'e-mail. C'est l'occasion ici de vous montrer comment crypter une donnée. Le fichier comporte des fonctions de cryptage et de décryptage. Il sera inclus en haut des fichier qui demandent ce type de service.

Créons maintenant le fichier *chiffrer.inc.php* qui sera inclus en haut de chaque fichier usant du cryptage de chiffres :

Listing 6.6 : *chiffrage : fonctions.inc.php*

```
global $serrure=85465465454;
//changez la serrure de temps en temps
function crypte_chiffre($secret){
$secret*=$serrure;
$secret=base64_encode($secret);
$secret=urlencode($secret);
return $secret;
}
function decrypte_chiffre($revelation){
$revelation =base64_decode($revelation);
$revelation /=$serrure;
return $revelation;
}
```

Du côté de la création de la donnée, nous obtiendrons :

```
include "../commun/chiffrer.inc.php";
```

Puis avant d'écrire le lien dans l'e-mail :

```
$clef= crypte_chiffre($clef);
```

Enfin, ajoutez le lien dans l'e-mail :

```
echo "Pour se désinscrire du club des Bons Vivants cliquez sur ce
lien : http://www.bonsvivants.com/club/demission.php?id=$clef ";
```

```
/*attention de laisser un espace blanc après le lien pour que
 le logiciel de messagerie puisse identifier ce texte comme un lien
  hypertexte.*/
```

En haut du fichier *demission.php*, ajoutez du code qui sert à vérifier la nature de la donnée envoyée :

```
if (!$_GET]){
header("Location: http://www.bonsvivants.com/");
exit;
}
if ($_GET['id']){
$clef=decrypte_chiffres($_GET['id']);
}
$sql="select clef from membres where clef='$clef'";
$resultat=mysql_query($sql,$id_link);
$nombre =mysql_num_rows($resultat);
if (!$nombre || $nombre!=1){
header("Location: http://www.bonsvivants.com/");
exit;
}
```

Vous n'utiliserez peut-être pas ces fonctions et vous contenterez de mettre un lien vers le tableau de bord du club :

```
RACINEWWW."/club/index.php?nom_usage=$nom_usage
&mot_passe=$mot_passe";
```

Le nettoyage des adresses e-mail obsolètes

Une fois votre courriel envoyé, vous constaterez que certaines adresses e-mail sont obsolètes. La solution serait de les enlever directement de la table *membres* ou de prévoir un délai d'un mois pour qu'un membre étourdi puisse modifier son adresse dans le club. Pour cela, nous créons la table *emails_perimes* en entrant cette requête dans la fenêtre de PhpMyAdmin afin de lancer les commandes :

```
CREATE TABLE emails_perimes (email VARCHAR(75) not null, date_entree
TIMESTAMP(8) not null , PRIMARY KEY (email), INDEX (email), UNIQUE
(email))
```

Passons au formulaire qui sert à entrer les adresses e-mail périmées dans la table :

```
<?
if ($_POST['email']){
$sql="insert into emails_perimes (email) VALUES('$email')";
mysql_query($sql,$id_link);
}
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
 <title>Entrée des adresses emails périmées</title>
</head>
<body>
<?
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
?>
email périmé :<input type="text" name="email">
<input type="submit" value="Entrer">
</form>
```

Pour fixer un délai, nous proposerons une fenêtre dans le tableau de bord du membre qui permettra à ce dernier d'entrer sa nouvelle adresse e-mail. Si son adresse n'est pas modifiée dans ce délai, le membre sera radié du club.

Pour épurer les tables *membres* et *emails_perimes*, nous utiliserons la jointure externe. Nous mettrons ce morceau de code dans le fichier *votre_email.php*, étudié plus loin. Il vaut mieux faire ces deux opérations de changement d'e-mail et d'épuration de la table *emails_perimes* dans le même temps, sinon il se pourrait que le membre revienne sur la page et qu'on lui demande encore sa nouvelle adresse e-mail.

La jointure externe (`LEFT [OUTER] JOIN`) permet de comparer la valeur des lignes de deux colonnes de deux tables différentes. Si les deux valeurs ne sont pas identiques, le résultat retourne la valeur `NULL`. Ce qui nous intéresse ici, c'est d'effacer de la table *emails_perimes* les adresses e-mail obsolètes, une fois qu'elles ont pu être modifiées. Ces adresses e-mail sont dans la table *emails_perimes* mais n'apparaissent plus dans la table *membres*, puisqu'elles ont été modifiées par leur possesseur. Dans une jointure externe, ces adresses e-mail seront retournées comme `NULL`.

Nous effaçons ensuite ces résultats `NULL` de la table *emails_perimes*. La syntaxe de la jointure externe est `LEFT OUTER JOIN nom_table ON`. Le

mot OUTER n'est pas nécessaire. Dans la première requête, nous sélectionnons les données et effaçons les lignes de la table *emails_perimes* qui retournent NULL. Puis nous sélectionnons les adresses qui datent de plus d'un mois dans la table *emails_perimes* et effaçons les lignes contenant ces adresses dans les deux tables.

```
/////ENLÈVE LES EMAILS RECTIFIÉS DE LA LISTE DES EMAILS PÉRIMÉS//
$sql="select emails_perimes.* from emails_perimes
 LEFT JOIN membres ON emails_perimes.email=membres.email
 where membres.clef IS NULL";
$resultat=@mysql_query($sql,$id_link);
while($rang=@mysql_fetch_array($resultat)){
$email=$rang['email'];
$sql_efface="delete from emails_perimes WHERE email='$email'";
@mysql_query($sql_efface,$id_link);
}
$hier=date ("Ymd", mktime (0,0,0,date("m")-1,date("d"),date("Y")));
 //il y a un mois
$sql ="select email from emails_perimes WHERE moment<$hier";
$resultat=@mysql_query($sql,$id_link);
 while($rang=@mysql_fetch_array($result)){
 $email=$rang['email'];
 $sql_efface="delete from emails_perimes where email='$email'";
 @mysql_query($sql_efface,$id_link);
 $sql_efface="delete from membres where email='$email'";
 @mysql_query($sql_efface,$id_link);
 }
```

6.4 Des aménagements dans l'application

Récapitulons les modifications sur l'application au vu des dernières applications créées.

Vous allez créer un fichier nommé *authentification.inc.php* que vous inclurez dans tous les fichiers demandant une authentification.

Dans le tableau de bord des membres, nous ajouterons une requête qui vérifiera si l'adresse e-mail est contenu dans la table *emails_perimes*. Dans ce cas, une fenêtre s'ouvrira pour proposer au membre du club de saisir la nouvelle.

Le fichier *authentification.inc.php* fournit les valeurs de `membres.clef`, `email` et `nom_usage` :

```
$sql="select * from emails_perimes where email='$email'";
$resultat=@mysql_query($sql,$id_link);
$nbre_rangs=mysql_num_rows($resultat);
if ($nbre_rangs==1){
$fenetre=1;
}
```

Dans l'en-tête, ajoutez un script en langage Javascript :

```
if ($fenetre){
echo "function ouvrir_vasistas(adresse,nom_fenêtre,caracteristiques)
 {\n";
echo "window.open(adresse, nom_fenêtre,caracteristiques);\n";
echo '}';
}
```

Affichez enfin la balise <BODY> :

```
<?
if ($fenetre){
echo "<body onLoad=\"ouvrir_vasistas('votre_email.php','vasistas',
'menubar=no,scrollbars=no,statusbar=no,width=300,height=200')\">\n";
}
else {
echo "<body>";
}
?>
```

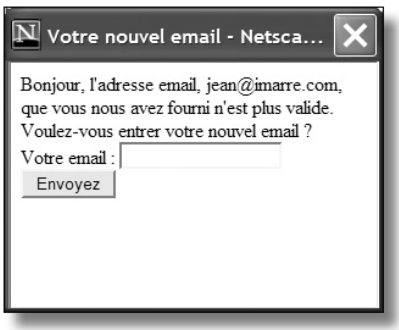
Nous utilisons l'événement `onLoad` qui se déclenche au chargement de la page dans le navigateur. Le script Javascript est créé par la commande `echo`, liée à la condition de l'existence de l'e-mail dans la table *emails_perimes*.

Le fichier *votre_email.php* est simple :

Listing 6.7 : *modification de l'email votre_email.php*

```
<?
session_start() ;
$clef=$_SESSION['clef'] ;
if ($_POST["nouvel_email"]){
$nouvel_email=$_POST["nouvel_email"] ;
$sql="update membres set email='$nouvel_email' where clef='$clef'";
```

```
mysql_query($sql,$id_link);
}
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
 <title>Votre nouvel email</title>
</head>
<body>
<?
if (!$nouvel_email){
echo "Bonjour, l'adresse email, $email, que vous nous avez fournie
 n'est plus valide. Voulez-vous entrer votre nouvelle adresse ?";
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
echo 'Votre email : <input type="text"
name="nouvel_email">';
echo '<input type="submit" value="Envoyez">';
echo '</form>';
}
else {
echo "Merci de votre contribution votre adresse email a été changée
 en $nouvel_email";
]
?>
</body>
</html>
```



◄ Figure 6.7 : *Votre e-mail est périmé*

L'affichage de ce fichier est l'émanation d'un code Javascript, il n'est donc pas possible de rappeler d'autres en-têtes avec les commandes `include_once` habituelles sans que le moteur PHP ne proteste.

6.5 En-têtes HTTP et variables globales

Elles comprennent les requêtes HTTP, les variables fournies par le serveur et nourries automatiquement par le navigateur de l'internaute et les tableaux créés par le moteur PHP pour conserver les requêtes CGI pour les formulaires (`POST`, `GET`…). Il est indispensable de les connaître car elles participent constamment à l'interactivité de vos programmes.

En-têtes HTTP

Voici les fonctions qui vous permettent d'interagir avec les en-têtes HTTP, en y lisant, en y écrivant des valeurs ou en les effaçant.

Tab. 6.1 : Les fonctions pour les en-têtes HTTP

Fonction	Description
`int setcookie ( string nom [, string valeur [, int expire [, string chemin [, string domaine [, int secure]]]]])`	Envoie un cookie sur la machine client.
`bool session_destroy ( void)`	Efface les données enregistrées dans la session.
`string session_id ( [string id])`	Retourne le nom (`id`) de la session courante et, si un `id` est ajouté en paramètre, change l'`id` courant par celui-ci.
`bool session_is_registered ( string nom)`	Vrai si une variable est enregistrée dans une session
`string session_name ( [string nom])`	Retourne ou change le nom de la session courante.
`bool session_register ( mixed nom [, mixed ...])`	Enregistre une ou plusieurs variables dans la session courante.
`bool session_unregister ( string nom)`	Efface une variable de la session courante.
`string session_save_path ( [string chemin])`	Retourne ou change le chemin de la session courante.

Tab. 6.1 : Les fonctions pour les en-têtes HTTP

Fonction	Description
`bool session_start ( void)`	Crée une session si aucune session n'est active ou reprend la session existante en initialisant les variables.
`int header ( string chaîne [, bool remplace])`	Envoie un en-tête HTTP. Si `remplace` est sur `FALSE`, votre en-tête ne remplacera pas un en-tête de même type.
`array getallheaders ( void)`	Retourne un tableau de tous les en-têtes `HTTP` passés à la page.

Ces fonctions doivent impérativement être placées au début de votre script, avant la balise <HTML>.

Quelques utilisations de la fonction header()

Vous pouvez rediriger un navigateur sur une autre page avec le paramètre `location` qui est un équivalent de la balise HTML `refresh` :

```
header("Location: http://www.ebizous.com/");
```

URL absolue

L'URL de redirection doit être absolue. Si votre URL est un chemin relatif, la redirection n'aura pas lieu.

Plusieurs noms de domaine sur le même site

Certains hébergeurs vous permettent gratuitement de pointer plusieurs noms de domaine sur le même site. Il vous suffit alors de mettre quelques lignes sur votre page d'accueil pour rediriger le visiteur sur la page choisie selon l'URL demandée, cela d'une manière transparente.

```
if (eregi($_SERVER["HTTP_HOST"],"votreautresite.com")){
echo '<meta http-equiv="refresh"
 content="0;URL=http://monsite.com/repertoire/">';
exit;
}
```

Il est conseillé de mettre le deuxième site, si c'est un site différent, dans un répertoire qui lui est propre, et de créer des fichiers de redirection dans le répertoire principal pour toutes les pages du deuxième site afin d'éviter de taper à chaque fois le nom du répertoire. N'oubliez pas la fonction `exit`, sinon vous afficherez les deux pages. Vous ne pouvez ici utiliser la fonction `header()` car aucune ligne ne doit la précéder.

Les fournisseurs d'accès utilisent les `proxys`. Ces machines stockent les pages les plus fréquemment vues par leurs internautes. Ce procédé évite de payer de la bande passante mais se révèle gênant pour le webmestre d'un site, car la page sera mise à jour seulement de temps en temps. D'autre part, le navigateur, si certaines options sont laissées par défaut, peut enregistrer les pages sur le disque dur. Pour peu que vous affichiez la date et/ou de l'heure, ou mettiez fréquemment à jour votre site, l'internaute pensera que votre site fonctionne mal ou n'est pas mis à jour. Voici des lignes de programmes qui, envoyés dans la fonction `header()`, obligeront le `proxy` à recharger la page à chaque visite.

```
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . " GMT");
```

Ici la page envoie une information indiquant qu'elle est constamment modifiée.

```
header("Pragma: no-cache");
```

ici elle empêche l'ordinateur de mettre la page dans son cache ou le proxy de la stocker.

Si votre site est multilingue, vous pouvez afficher des liens sur des drapeaux à la page d'accueil, mais pour le confort des visiteurs, vous pouvez aussi rediriger l'internaute vers la page adéquate selon la langue choisie dans le navigateur. Ainsi ils court-circuitent la page d'accueil qui affiche un lien par langue et n'a, en général, pas d'intérêt.

```
if (eregi($_SERVER["HTTP_ACCEPT_LANGUAGE"],"fr")){
echo '<meta http-equiv="refresh"
content="0;URL=http://www.bonsvivants.com/fr/">';
exit;
}
```

Ajoutez ensuite des blocs `elseif` pour les langues que vous proposez, jusqu'à un `else` qui oriente le visiteur vers la page d'accueil, si son navigateur n'a été réglé pour aucune des langues que vous proposez.

Variables PHP

Le moteur PHP dispose de variables globales qui complètent les variables Apache (ou variables d'environnement) que nous avons vues dans le chapitre 4.

Tab. 6.2 : Les variables PHP

Variable	Description
`$PHP_SELF`	Équivalent à `SCRIPT_NAME`
`$HTTP_COOKIE_VARS`	Tableau des variables reçues d'un cookie
`$HTTP_GET_VARS`	Tableau des variables reçues d'un formulaire par la méthode `GET`
`$HTTP_POST_VARS`	Tableau des variables reçues d'un formulaire par la méthode `POST`
`$HTTP_POST_FILES`	Tableau des variables reçues d'un formulaire de téléchargement de fichiers
`$HTTP_ENV_VARS`	Tableau des variables d'environnement
`$HTTP_SERVER_VARS`	Tableau des variables reçues d'un cookie
`$HTTP_SESSION_VARS`	Tableau des variables reçues d'une session

Voici les nouvelles variables fournies avec la version 4 de PHP :

Tab. 6.3 : Les variables PHP (PHP4)

Variable	Description
`$_COOKIE`	Tableau des variables globales reçues d'un cookie
`$_GET`	Tableau des variables globales reçues d'un formulaire par la méthode `get`
`$_POST`	Tableau des variables globales reçues d'un formulaire par la méthode `post`

Tab. 6.3 : Les variables PHP (PHP4)

Variable	Description
`$_FILES`	Tableau des variables globales reçues d'un formulaire de téléchargement de fichiers
`$_ENV`	Tableau des variables globales d'environnement
`$_SERVER`	Tableau des variables globales reçues d'un cookie
`$_SESSION`	Tableau des variables globales reçues d'une session
`$_REQUEST`	Tableau né de la fusion des tableaux `$_GET`, `$_POST` et `$_COOKIE`

Ces variables servent de raccourcis de code, de marqueurs ou de variables génériques pour tout type d'environnement.

Il est plus simple de taper `$_SERVER["PHP_SELF"]` dans l'action d'un formulaire pour que l'application se rappelle elle-même plutôt qu'elle n'appelle le chemin relatif ou absolu au fichier. Vous pouvez déplacer votre fichier dans un autre répertoire ou le placer sur un autre serveur sans le changer.

Nous avons vu aussi que vous pouvez tester la présence du tableau `$_GET` pour savoir si un lien a été cliqué avec les données qui s'y rattachent. Avec `$_POST`, nous avons créé une application qui traite les données de toutes sortes de formulaires.

Voyez le traitement générique d'un formulaire quelconque dans le chapitre *Les clefs du PHP*.

Variable d'environnement

Les variables d'environnement sont les variables du serveur, en l'occurrence Apache. Le navigateur du visiteur ne fournit pas forcément toutes celles qui lui sont imparties. Ainsi, essayez en stockant la variable `$_SERVER["HTTP_REFERER"]` de vos visiteurs. C'est une variable intéressante car elle donne l'URL de la page qui a appelé la page courante. Ainsi, vous verrez les sites qui ont placé des liens vers votre site, donc

ceux qui attirent le plus de monde chez vous. Vous remarquerez aussi que parfois, la page ne contient aucune valeur. Avec la version 4, `$HTTP_ACCEPT` est remplacé par `$_SERVER` comme `$_SERVER["HTTP_ACCEPT"]`.

Tab. 6.4 : Les variables d'environnement

Variable	Description
`$HTTP_ACCEPT`	Liste des formats acceptés par le navigateur web
`$HTTP_ACCEPT_LANGUAGE`	Langage accepté par le navigateur
`$HTTP_CONNECTION`	Numéro d'IP et numéro de port éventuel
`$HTTP_HOST`	Numéro d'IP et numéro de port éventuel
`$HTTP__REFERER`	Page d'origine de la requête
`$HTTP_USER_AGENT`	Version du navigateur
`$REMOTE_ADDR`	Adresse IP de l'internaute
`$REMOTE_PORT`	Port client
`$REQUEST_METHOD`	Méthode GET ou POST
`$SERVER_PROTOCOL`	Nom du protocole utilisé par le serveur (HTTP/1.0)
`$SERVER_SOFTWARE`	Nom du serveur HTTP (Apache/1.3.20 (Win32) PHP/4.0.6)

◄ Figure 6.8 : *Les variables d'environnement*

6.6 Récapitulation

Créons un fichier *variables.inc.php* dans lequel nous stockons les constantes et les variables prédéfinies comme celles concernant le jour courant… Au début de vos fichiers, vous obtiendrez une liste de commande `include` que vous modulerez avec les deux barres obliques caractérisant les commentaires. Ainsi vous neutraliserez la ligne concernant l'authentification pour les fichiers hors du club. Dans votre répertoire *administration*, vous n'aurez pas besoin d'authentification ni de statistiques. Au début de vos fichiers, vous obtiendrez à peu près :

```
include "../commun/variables.inc.php";
include RACINE."/commun/fonctions.inc.php";
include RACINE."/commun/connexion.inc.php";
//include RACINE."/commun/authentification.inc.php";
//include RACINE."/commun/stats.inc.php";
```

Il sera sans doute mieux pour vous d'avoir une liste imprimée des fonctions contenues dans le fichier *fonctions.inc.php* avec leurs fonctionnalités et paramètres. Imprimez également la liste des variables et des constantes contenues dans le fichier *variables.inc.php*, ainsi que la liste des variables extraites par l'authentification dans la table *membres*.

7

Des algorithmes, des outils et des fonctions

Lorsque vous connaissez les fonctions principales et savez les articuler dans un programme, vous pouvez résoudre n'importe quel problème, à condition de bien le poser au départ. Avant de créer une application, que ce soit par passion ou dans le cadre de votre travail, pensez-y plusieurs jours durant, compulsez des documents sur Internet, validez les différentes méthodes et choisissez enfin celle qui apporte un juste équilibre entre la difficulté et la simplicité. Une application trop simple peut se révéler fastidieuse, mais une application trop complexe demande parfois beaucoup d'énergie pour un détail qui se révèle finalement inutile. Si d'autres personnes ont déjà résolu votre problème, totalement ou partiellement, proposez-leur des améliorations. Si le programme n'existe pas mais qu'il vous dépasse par son ampleur, élaborez-le en partie. Puis rédigez une documentation en détaillant vos objectifs et les étapes à venir. Une fois le programme finalisé, proposez-le sur votre site personnel ou sur un site touchant la communauté du Libre.

Dans le chapitre **Mettre de l'ordre**, nous vous donnons des clés pour simplifier votre application et créer un fichier de configuration qui regroupe toutes les variantes de l'application.

Nous vous proposerons quelques outils pour ajouter à votre site. Mais, avant cela, nous nous pencherons sur les tableaux car la plupart des applications les utilisent, y compris dans leurs fonctions de tri.

7.1 Tableaux

Quand vous avez plusieurs résultats de SQL et avez besoin de les ordonner, les tableaux deviennent indispensables.

Tab. 7.1 : Les fonctions de manipulation d'éléments de tableau

Fonction	Description
`array array(…)`	Crée un tableau à partir d'une liste de valeurs.
`array array_keys(array tableau1)`	Crée un tableau avec les clés de l'autre tableau.
`array array_merge (array tableau1, tableau2…)`	Fusionne deux tableaux ou plus. Pour deux clés identiques, il gardera seulement la dernière valeur.

Tab. 7.1 : Les fonctions de manipulation d'éléments de tableau

Fonction	Description
`array array_merge_recursive (array tableau1, tableau2…)`	Fusionne deux tableaux. En cas de clés identiques, insère un tableau au lieu d'une variable scalaire.
`array array_pop (array tableau1)`	Retourne la dernière valeur, qu'il supprime du tableau.
`array array_push (array tableau1, mixed var,…)`	Ajoute un ou plusieurs éléments à la fin du tableau.
`array array_shift (array tableau1)`	Extrait la première valeur, qu'il supprime du tableau.
`array array_slice (array tableau1, int debut, int [longueur])`	Supprime une partie des éléments du tableau.
`array array_splice (array tableau, int début, int [longueur], array tableau2)`	Supprime une partie des éléments du tableau ou les remplace par les éléments du deuxième tableau.
`array array_unshift (array tableau1, mixed var,…)`	Ajoute un ou plusieurs éléments au début du tableau.
`array array_diff ( array tableau1, array tableau2 [, array ...])`	Crée un tableau avec les éléments du tableau1 qui ne sont pas dans le tableau2, etc.
`array array_intersect ( array tableau1, array tableau2 [, array ...])`	Crée un tableau avec les éléments communs de tous les tableaux.
`array array_values(array tableau1)`	Crée un tableau avec les clés de l'autre tableau.
`int array_walk ( array tableau1, string fonct [, mixed parametre])`	Applique à chaque élément du tableau une fonction pour laquelle il est possible d'envoyer un troisième paramètre.
`array range (int bas, int haut)`	Crée un tableau formé d'un intervalle de valeurs.
`array compact(string var [, array, tableau…])`	Crée un tableau associatif formé des noms et valeurs de variables (éventuellement contenues dans un tableau).

Tab. 7.1 : Les fonctions de manipulation d'éléments de tableau

Fonction	Description
`int extract ( array tableau [, int type_extract [, string prefixe]])`	Crée une variable avec la paire clé/valeur de chaque élément du tableau, les deuxième et troisième paramètres sont des options.
`int sizeof (array tableau)` `int count (array tableau)`	Retourne le nombre d'éléments du tableau, 1 pour une variable scalaire.
`array each (array tableau)`	Retourne la prochaine paire clé/valeur du tableau.
`void list(….)`	Affecte respectivement des valeurs à une liste de variables.
`boolean in_array(mixed aiguille, array bottedefoin)`	Retourne `vrai` si le tableau contient la valeur.
`mixed key (array tableau)`	Retourne la clé de l'élément courant.

Types d'extraction

La fonction `extract()`, qui crée des variables à partir d'un tableau, utilise des options dans le cas où deux éléments identiques du tableau des symboles provoquent une collision de variables. Le préfixe est suivi d'un blanc souligné (_). Le tableau des symboles, créé automatiquement par PHP, est le tableau qui contient toutes les variables.

Tab. 7.2 : Les options d'extraction d'éléments de tableau

Type	Effet
`EXTR_OVERWRITE`	Par défaut. En cas de collision, écrase la variable précédente.
`EXTR_SKIP`	En cas de collision, épargne la variable précédente.
`EXTR_PREFIX_SAME`	En cas de collision, préfixe le nom de la variable avec le troisième paramètre, conservant ainsi l'ancienne variable.
`EXTR_PREFIX_ALL`	Préfixe toutes les variables. Depuis PHP 4.0.5, cela inclut les noms numériques de variables, qui autrement seraient invalides.
`EXTR_PREFIX_INVALID`	Seuls les noms invalides ou numériques de variables sont préfixés. Ce drapeau a été ajouté à la version 4.0.5.

Tab. 7.2 : Les options d'extraction d'éléments de tableau

Type	Effet
`EXTR_IF_EXISTS`	En cas de collision, écrase seulement la variable précédente, sinon ne fait rien. Ce drapeau a été ajouté à la version 4.2.0.
`EXTR_PREFIX_IF_EXISTS`	En cas de collision, ajoute un préfixe seulement si la variable non préfixée existe. Ce drapeau a été ajouté à la version 4.2.0.

Imaginez que vous parcouriez les pages d'un livre avec le doigt, ce doigt serait le pointeur. PHP se sert d'un pointeur pour parcourir un tableau. Au moment où un élément de tableau s'affiche, le pointeur le désigne.

Les fonctions suivantes manient le pointeur dans un tableau.

Le pointeur

Tab. 7.3 : Les fonctions de localisation du pointeur

Fonction	Description
`Mixed end (array tableau)`	Place le pointeur interne sur le dernier élément du tableau.
`Mixed next (array tableau)`	Place le pointeur interne sur l'élément suivant.
`Mixed pos (array tableau)`	Retourne la position courante du pointeur.
`Mixed prev (array tableau)`	Place le pointeur avant la position courante.
`Mixed reset (array tableau)`	Place le pointeur au début du tableau.

Le tri

Les fonctions de tri ne retournent pas de valeur.

Voici les fonctions de tri simple.

Tab. 7.4 : Les fonctions de tri simple

Fonction	Description
`Void sort (array tableau)`	Trie le tableau.
`Void rsort (array tableau)`	Trie le tableau dans l'ordre inverse.
`Void uksort (array tableau, function fonction)`	Trie le tableau par clés en utilisant une fonction.
`Void usort (array tableau, function fonction)`	Trie le tableau par valeurs en utilisant une fonction.
`Void shuffle (array tableau)`	Mélange aléatoirement les éléments du tableau.

Voici les fonctions de tri associatif, c'est-à-dire celles qui conservent l'association d'origine entre les index et les valeurs.

Tab. 7.5 : Les fonctions de tri associatif

Fonction	Description
`Void asort (array tableau)`	Trie les éléments dans l'ordre alphabétique inversé.
`Void arsort (array tableau)`	Trie les éléments dans l'ordre alphabétique.
`Void ksort (array tableau)`	Trie le tableau dans l'ordre de ses clés.
`Void krsort (array tableau)`	Trie le tableau dans l'ordre inverse de ses clés.
`Void uasort (array tableau, function fonction)`	Trie le tableau en utilisant une fonction.
`Void sort (array tableau)`	Trie le tableau dans l'ordre de ses valeurs.

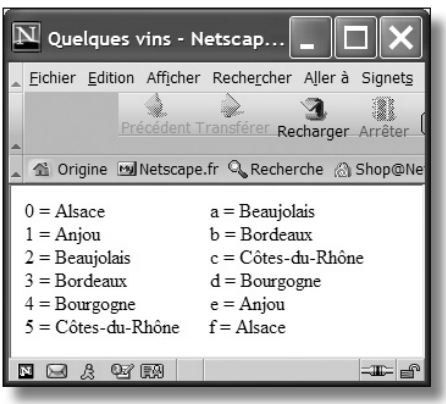
Les tableaux servent à classer des données et à les manipuler. Vous pouvez le trier ou le modifier en enlevant, en ajoutant ou en intervertissant des valeurs, fusionner plusieurs tableaux et en décliner chaque valeur dans une boucle `foreach`. Le tableau est l'outil favori de "l'analyste-jongleur" de données.

Déclinons un exemple de tri simple puis le même, de type associatif :

Listing 7.1 : *Tri simple par ordre alphabétique, tri associatif par clef :* **vins.php**

```
$vins = array ("Bourgogne", "Beaujolais", "Bordeaux",
 "Côtes-du-Rhône", "Alsace", "Anjou");
sort ($vins);
foreach($vins as $clef=>$valeur){
echo "$clef = $valeur <BR>\n";
}
echo "<P>";
$vins = array ("d"=>"Bourgogne", "a"=>"Beaujolais", "b"=>"Bordeaux",
 "c"=>"Côtes-du-Rhône", "f"=> "Alsace", "e"=>"Anjou");
ksort ($vins);
foreach($vins as $clef=>$valeur){
echo "$clef = $valeur <BR>\n ";
}
```

Ces deux exemples vous montrent la différence entre un tri simple et un tri associatif.



◄ Figure 7.1 : *Tris divers*

Comme vous le voyez ci-dessous, le tri simple attribue de nouvelles clés en fonction de la place de l'élément dans le tri, tandis que le tri associatif conserve les clés, se contentant d'afficher le nouvel ordre des éléments.

Résolvons maintenant le problème du tri alphabétique français. Si nous utilisons les fonctions de PHP, les mots commençant par une lettre

accentuée seront relégués après les mots en z. Pour remédier à ce problème, utilisons la fonction `usort()`, qui permet de faire un tri par une fonction :

```
function cmp ($a, $b) {
   $un_tableau=array('à','â','ä');
   $i=0;
   $lalettre='b';
   foreach($un_tableau as $valeur){
      $a=str_replace($valeur,$lalettre.$i,$a);
      $b=str_replace($valeur,$lalettre.$i,$b);
      $i++;
   }
```

Voici le vrai corps de la fonction. Les lignes qui suivent ne font que répéter le même schéma pour chaque lettre accentuée, à savoir que chaque lettre accentuée est placée après sa lettre générique et avant la lettre suivante en insérant un chiffre :

Listing 7.2 : *Tri avec les lettres accentuées : tri_franc.php*

```
   $un_tableau=array('é','è','ê','ë');
   $lalettre='f';
   $i=0;
   foreach($un_tableau as $valeur){
      $a=str_replace($valeur,$lalettre.$i,$a);
      $b=str_replace($valeur,$lalettre.$i,$b);
      $i++;
   }
   $un_tableau==array('ô','ö');
   $lalettre='p';
   $i=0;
   foreach($un_tableau as $valeur){
      $a=str_replace($valeur,$lalettre.$i,$a);
      $b=str_replace($valeur,$lalettre.$i,$b);
      $i++;
   }
   $un_tableau=array('î','ï');
   $i=0;
   $lalettre='j';
   foreach($un_tableau as $valeur){
      $a=str_replace($valeur,$lalettre.$i,$a);
      $b=str_replace($valeur,$lalettre.$i,$b);
      $i++;
   }
   $un_tableau=array('ç');
   $i=0;
   $lalettre='d';
```

```
   foreach($un_tableau as $valeur){
      $a=str_replace($valeur,$lalettre.$i,$a);
      $b=str_replace($valeur,$lalettre.$i,$b);
      $i++;
   }
   $un_tableau=array('ù','û');
   $i=0;
   $lalettre='v';
   foreach($un_tableau as $valeur){
      $a=str_replace($valeur,$lalettre.$i,$a);
      $b=str_replace($valeur,$lalettre.$i,$b);
      $i++;
   }

   if ($a == $b) return 0;
   return ($a < $b) ? -1 : 1;
}
```

Cette fonction s'applique à chaque élément du tableau. Ici, nous utilisons une astuce qui consiste à remplacer la lettre accentuée par la lettre alphabétique suivante et un chiffre. Un chiffre sera toujours trié avant une lettre. Ce nouveau mot donnera la clé du vrai mot dans le tri du tableau.

▲ Figure 7.2 : *Tri français*

Pour appeler cette fonction, il vous reste à écrire la ligne suivante :

```
usort ($tableau, "cmp");
```

7.2 Calendriers, dates et autres agendas

Les fonctions de date sont des outils pour nettoyer la base, trier des données, connaître l'actualité d'une donnée et aménager des délais avant une action… Leurs fonctionnalités sont innombrables.

Durée après une date ou avant une date

Nous avons déjà vu comment afficher une date. Vous pouvez avoir envie de faire un concours ou un sondage avec un lot à gagner, fourni par un sponsor en échange de publicité, comme nous le verrons plus loin. Pour inciter les visiteurs à y participer, vous afficherez une date limite.

Pour une date dans le passé ou le futur, vous pouvez utiliser la fonction `date()`. Celle-ci prend deux arguments. Le premier est le format et le second, optionnel, est le `timestamp` d'Unix (nombre de secondes depuis le 1er janvier 1970). Si ce second argument n'est pas fourni, la fonction utilise par défaut le `timestamp` du moment où la fonction est appelée.

`Date("n")` vous donne le mois courant sans adjonction de 0 et `date("Ymd")` vous retourne une date dans le format *aaaammjj*.

La fonction `mktime()` crée un `timestamp` à partir des arguments respectifs : heure, minute, seconde, mois, jour, année. Cette fonction est intéressante dans la mesure où elle rectifie les approximations de date ; ainsi, alors que le mois de février comprend 28 jours en 2003, `mktime(0,0,0,30,2,2003)` donnera le `timestamp` du 2 mars 2003.

Pour afficher une date quinze jours auparavant, écrivez votre fonction comme suit :

```
echo date("j/n/Y", mktime(0,0,0,date("n"), date("j")-15, date("Y")));
```

Pour faire le décompte avec une date dans le futur, comme la date limite pour gagner un concours :

```
$date_butoir=mktime(0,0,0,9, 26, 2002);
/*ici vous avez le timestamp de la date butoir qui est
 le 26 septembre 2002*/
$jours_difference=(round(($date_butoir-time())/(60*60*24)));
echo $jours_difference;
```

Vous décomptez le nombre de secondes de la date avec le nombre de secondes du moment présent. Vous divisez pour avoir des jours, puis faites un arrondi du résultat pour éviter les virgules. Vous avez à chaque moment le nombre de jours qui séparent les visiteurs d'un gain éventuel.

Pour une date dans le passé, c'est le même principe.

Affichage d'un calendrier

Pour afficher un calendrier, qui n'est rien moins qu'un tableau HTML, utilisez la fonction `date()`.

Un calendrier qui donne le mois courant est un tableau composé de six rangs de sept cases. Le premier rang donne le titre de chaque colonne : dimanche, lundi, mardi, mercredi, jeudi, vendredi, samedi.

Ensuite, le mois peut commencer un jour ou un autre de la semaine. Le jour courant sera signalé par une couleur différente. Un formulaire doté d'un menu permet d'afficher n'importe quel mois.

Pour ce faire, nous tisserons notre tableau avec une boucle qui s'arrêtera au nombre de jours dans le mois : `date("t")`.

```
for ($=1;$i>=date("t");$i++)
```

Pour le premier rang (en fait, le rang en dessous du nom des jours), on utilise :

```
date("w", mktime(0,0,0,date("n"), 1, date("Y")));
```

La semaine commence le lundi, qui correspond au jour 1, et finit le dimanche, qui est le jour 0.

Commençons notre programme par le formulaire d'affichage. Il est composé d'un menu déroulant qui donne le choix sur six mois dans le passé et douze dans le futur. C'est une boucle.

Listing 7.3 : *Agenda sous forme de calendrier : agenda.php*

```
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
echo '<select name="instant" size="1">';
$date_inf= mktime(0,0,0,date("n")-6, date("j"), date("Y"));
$date_sup= mktime(0,0,0,date("n")+12, date("j"), date("Y"));
```

```
$mois_franc=array('', 'janvier', 'février', 'mars', 'avril', 'mai',
 'juin', 'juillet', 'août', 'septembre', 'octobre', 'novembre',
 'décembre');
for ($variable_mois=-6,$variable_tableau=date("n",$date_inf);
$variable_mois<=12;
$variable_mois++,$variable_tableau++){
if ($variable_tableau==13){
$variable_tableau=1;
}
$chronos=mktime(0,0,0,date("n")+$variable_mois, date("j"), date("Y"));
echo "<option value=\"$chronos\"";
if (!$variable_mois){
echo ' SELECTED';
}
$mois_courant=$mois_franc["$variable_tableau"];
$an_courant=date("Y", $chronos);
echo ">$mois_courant ${an_courant}</option>";
}
if (!$instant){
$instant=mktime(0,0,0,date("n"), date("j"), date("Y"));
}
$mois=date("n","$instant");
$mois_en_franc=$mois_franc["$mois"];
$annee =date("Y","$instant");
$taille_calendrier="40%";

?>
```

Dans ce formulaire, nous fabriquons la variable `$mois` avec le `timestamp` de chaque mois (six mois dans le passé et douze dans le futur). Si aucune sélection n'est choisie, le `timestamp` retenu par défaut est celui du jour courant.

La boucle produit deux variables indépendantes. Ici, `$variable_mois` est la variable qui parcourt tous les mois à partir d'un point de départ correspondant à six mois avant la date du jour et d'une limite incluse correspondant à douze mois après la date courante. La deuxième variable est `$variable_tableau`, qui parcourt le tableau des noms de mois en français. Le tableau commence par une cellule vide pour plus de confort ; ainsi, chaque mois correspond à son indice (janvier à 1, février à 2…). Le test sur `!$variable_mois` (ou `$variable_mois==0`) permet d'avoir le mois courant, qui sera le mois affiché par défaut dans le menu déroulant.

Nous finissons cette partie de programme en extrayant le mois et l'année de la variable `$instant` pour afficher le mois en français. Par ailleurs, nous alimentons une variable `$taille_calendrier` qui permettra de moduler la taille du tableau HTML.

Le tableau commence avec deux lignes, une pour afficher le mois et l'année, l'autre pour afficher les jours de la semaine :

```
echo "<table width=$taille_calendrier><TR><TD COLSPAN=\"7\">
$mois $annee</TD</TR>";
echo "<TR><TD>lundi</TD> <TD>mardi</TD> <TD>mercredi</TD> <TD>jeudi
</TD> <TD>vendredi</TD> <TD>samedi</TD> <TD>dimanche</TD></TR>";
```

Nous arrivons maintenant au cœur de l'algorithme du programme, c'est-à-dire à l'écriture de la boucle de création du tableau. Elle contient trois variables ; la première est la variable `$jours`, la deuxième, la variable `$jour_semaine`, et la troisième, `$cellules`, qui s'arrête à 35 quel que soit le nombre de jours. Le jour de semaine est un modulo de la variable `$cellules` divisé par 7 ; ainsi, cette variable `$jour_semaine` se réinitialise à 0 chaque dimanche, puisque le dimanche est toujours un multiple de 7. La variable `$jours` commence à s'afficher dès qu'elle est plus grande que 0. Pour l'initialiser à 1, il faut réunir des conditions.

Tab. 7.6 : Les conditions pour le tracé du tableau HTML et le comptage des jours du mois

Position du pointeur de la boucle	Condition
La première ligne du calendrier	`$cellules<8`
Le jour de la colonne correspondant au premier jour du mois	`$jour_semaine==date("w", mktime(0,0,0,$mois, 1, $annee))`

```
for ($jours=0,$cellules=1;$cellules<36; $cellules++){
$jour_semaine=$cellules%7;
if ($jour_semaine==1){
echo "<TR>";
}
/////////la ligne commence le lundi//////////////
if (date("j")==$jours && date("n")==$mois && date("Y")==$annee){
echo "<TD align=\"center\" bgcolor=\#FFFF66\">";
}
/*cette condition s'interroge si ce jour est aujourd'hui et
```

```
 dans ce cas la case est colorée*/
else {
echo "<TD align=\"center\">";
}
if ($cellules<8 && $jour_semaine==date("w", mktime(0,0,0,$mois, 1,
 $annee))){
$jours=1;
}
//cette condition initialise la variable jours//////////////
if ($jours>0 && $jours<=date("t",$instant)){
echo $jours;
}
else {
echo " ";
}
```

Le jour est affiché si la variable `jours` a été initialisée et si le nombre de jours dans le mois n'est pas dépassé. Si tel n'est pas le cas, on affiche le code HTML espace insécable (*Non Break Space*) pour que la bordure de la cellule soit affichée malgré tout.

```
echo "</TD>";
if ($jour_semaine==0){
echo "</TR>";
}
/////le dimanche finit la ligne/////////////
if ($jours>0){
$jours++;
}
if (date("t",$instant)==$jours && $jour_semaine==0){
break;
}
/*nous prévoyons le cas où un mois de février de 28 jours commence
 un lundi. Le break évite d'afficher une ligne de case vides*/
}
echo "<table>";
?>
```

Si vous voulez ajouter le saint du jour, vous trouverez, sur le site `http://www.chez.com/sarthe/liens_saints.html`, une liste de sites utiles.

Votre calendrier peut fonctionner en l'état, mais il serait plus intéressant d'y adjoindre un agenda, sans compter que vous devez également améliorer l'interface graphique.

Un agenda à jour

L'agenda se présentera sous deux formes : la première articulée sur l'affichage de l'agenda et la seconde sous la forme d'une liste. Pour entrer les données, il suffira de cliquer sur une icône dans le calendrier pour voir s'ouvrir une fenêtre et saisir du texte.

Les jours marqués d'un événement présentent une autre icône qui affiche l'événement dans une petite fenêtre. Nous pouvons y adjoindre l'envoi d'un courriel comme pense-bête.

Cette application sera à la disposition des membres inscrits dans le club. Nous étudierons l'application en éludant l'authentification, déjà évoquée dans le chapitre *L'internaute est d'abord une personne*.

La réflexion

Posez-vous les deux questions suivantes :

- Archive-t-on les informations obsolètes ?
- Envoie-t-on un courriel pour que la personne soit avertie de l'échéance d'un "jour à contenu" ?

Les choix sont de votre ressort, mais réfléchissez aux implications de chaque choix.

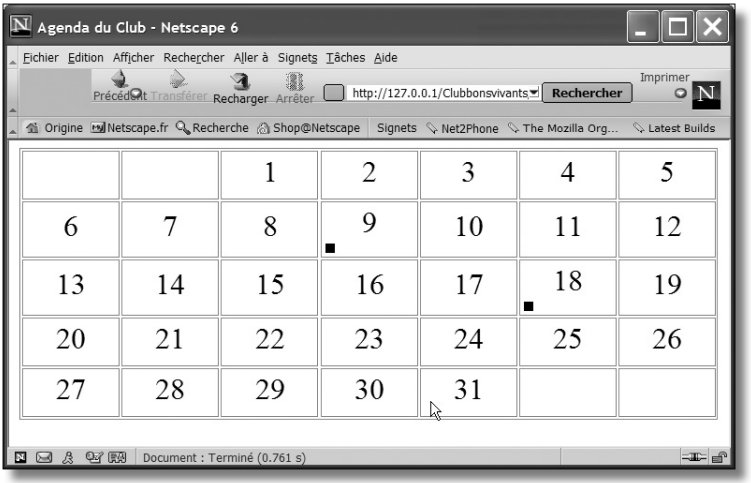
Parfois, nous nous plongeons dans d'anciens agendas. Il est agréable de se rappeler les types de personnes que nous rencontrions et d'activité que nous pratiquions alors. Le mieux est peut-être de laisser le choix aux membres du club. Au moment où ils décident de bénéficier de ce service, l'application leur propose de garder leurs archives ou de les détruire après un mois, un trimestre ou un an. Cela implique l'ajout dans la table d'un champ qui indique le type de conservation.

L'affichage de l'agenda sous forme de calendrier

Reprenons le code du calendrier et ajoutons-y une requête SQL. La table *agenda* sera constituée d'une clé, du nom d'usage de la personne, du type de texte, du texte et de la date sous la forme *AAAAMMJJHHSS*.

```
$sql="CREATE TABLE agenda (clef INT(11) DEFAULT '0' not null
 AUTO_INCREMENT, nom_usage VARCHAR(25) not null,
 type CHAR(1) not null, contenu TEXT not null, datedujour char(14)
 not null ,
PRIMARY KEY (clef), INDEX (clef), UNIQUE (clef))";
@mysql_query($sql,$id_link);
```

Le type `text` reçoit un grand nombre de données de type texte qu'il entrepose en binaire mais qui restent affichables dans PhpMyAdmin.



▲ Figure 7.3 : *Le calendrier comme agenda*

En haut de la page d'affichage de l'agenda, nous irons chercher ces informations :

```
if (!$moischoisi){
$moischoisi=date("Ym");
/*ici est affecté le mois par défaut à la variable si aucun mois
 n'est reçu par la méthode GET*/
}
$sql="select * from agenda where nom_usage='$nom_usage' AND
 datedujour LIKE '${moischoisi}%' ORDER BY datedujour";
$resultat=@mysql_query($sql,$id_link);
```

Vous venez de rencontrer deux nouvelles notions représentées par `LIKE` et `%`. `LIKE` est un opérateur de comparaison comme le signe = mais il permet

d'utiliser des jokers comme % qui désignent une suite de caractères. La requête demande à rechercher tous les champs de la table *agenda* où le nom d'usage est `$nom_usage` et la date du jour commence par la date courante au format *AAMMMM*. Nous ferons quelques modifications car on peut imaginer que le membre consulte n'importe quel mois. Par ailleurs, le fichier d'authentification nous donnera à chaque fois le nom d'usage. La commande `ORDER` ordonnera le résultat avant de le recevoir.

Créons un tableau indexé par la clé et contenant la date du jour, qui comprend la date et l'heure sous la forme *AAAAMMJJHHMMSS* :

```
while ($rang=mysql_fetch_array($resultat)){
$datedujour=$rang['datedujour'];
//j'extraie le jour
$lejour=(int)substr($datedujour, 6, 2);
$clef=$rang['clef'];
$contenu=$rang['contenu'];
$mots_contenus=split(" ", $contenu,7);
/*Le tableau obtenu contient un maximum de 7 éléments chacun composé
 d'un mot mais dont le dernier contient le reste du texte*/
$contenu='';
/*nous vidons la variable $contenu dont le contenu est dans le
tableau $mots_contenus */
$i=1;
while($i<6){
$contenu.=$mots_contenus[$i];
$i++;
}
//reconstitution de $contenu
$type=$rang['type'];
$rv["$lejour"][]=$clef;
$rv["$lejour"][]=$type;
$rv["$lejour"][]=$contenu;
}
/* Le tableau $rv a comme clef le jour du rendez-vous et comme valeur
un tableau avec toutes les autres données*/
$armoire_clefs=array_keys($rv);
/* sur cette ligne, on crée un tableau des clefs, c'est-à-dire de
 tous les jours qui sont présents dans la table. Nous savons que
 seuls les jours annotés sont présents.*/
```

Vous pouvez créer une table avec tous les saints de l'année, de 1 à 366, avec deux champs, le jour et le saint du jour. Pour cela, créez ici un tableau pour les saints du mois. Il suffit de faire une requête sur la table *toussaint* en trouvant le premier jour et le dernier jour du mois avec cette fonction :

```
function limites_mois ($moischoisi){
$mois=(int)substr ($moischoisi, 4, 2);
$an= substr ($moischoisi, 0, 4);
$debut=mktime(12,0,0,$mois, 1, $an);
$fin= mktime(12,0,0,$mois, date("t",$debut), $an);
```

Commençons par dépecer la variable `moischoisi`, qui est composée de quatre premiers chiffres donnant l'année et de deux chiffres suivants pour le mois. Nous avons forcé le type pour `$mois` en ajoutant `(int)` devant ; ainsi, 02 devient 2. Ce n'est pas obligatoire. La fonction `mktime()` qui suit fonctionne aussi avec 02 comme mois. Nous calculons ensuite le `timestamp` du premier jour du mois et celui du dernier jour avec la fonction `mktime()`. Pour la variable `$fin`, nous ne connaissons pas le dernier jour du mois, qui peut aller du 28 au 31. Pour le savoir, nous nous aidons de la fonction `date` avec le format *"t"* qui donne le nombre de jours dans le mois sur le `timestamp $debut`.

```
$debut = date("z",$debut);

/*les jours commencent à 0 ce qui va bien avec la requête SQL et
 la commande LIMIT*/
$fin = date("z",$fin);
$limites[]=$debut;
$limites[]=$fin;
return $limites;
//ici on retourne plusieurs valeurs à l'intérieur d'un tableau
}
$limites=limites_mois($moischoisi);
$debut=$limites[0];
$fin=$limites[1];
//$sql="select saints from toussaint limit $debut, $fin";
//$resultat=@mysql_db_query($dbname,$sql,$id_link);
//$tous_les_saints[]=' ';
/*on crée la première case vide pour que le jour du mois corresponde
 à l'indice*/
//while ($rang=mysql_fetch_array($resultat)){
//$tous_les_saints[]=$rang['saint'];
//}
?>
```

Maintenant, à l'intérieur de chaque cellule du tableau créé pour afficher le calendrier, nous allons introduire un tableau pour afficher le jour et en dessous deux cases, l'une pour la présence d'un événement et l'autre pour un autre élément de votre choix, par exemple le saint du jour.

Pour cela, nous insérons dans la cellule encadrée par la balise <TD> :

```
<TABLE><TR><TD colspan="2"></TD></TR><TR><TD>
</TD><TD></TD></TR><TABLE>
```

Voici maintenant le programme :

```
<div align="center"><TABLE width="95%« border=1>
<?
for ($jours=0,$cellules=1;$cellules<36; $cellules++){
$jour_semaine=$cellules%7;
```

Pour obtenir le jour de semaine, il suffit de faire le modulo de 7 :

```
if ($jour_semaine==1){
echo »<TR height=\"100\">";
}
```

Si le modulo est de 1, on commence la semaine par le lundi :

```
/////////la ligne commence le lundi//////////////
if (date("j")==$jours && date("n")==$mois && date("Y")==$annee){
echo "<TD align=\"center\" bgcolor=\#000000\">";
}
/*cette condition précédente s'interroge si ce jour est aujourd'hui
 et dans ce cas la case est colorée*/
else {
echo "<TD align=\"center\" valign=\top\">";
}
echo "<TABLE width=\"100\" border=0><TR height=\"50\">
<TD colspan=\"2\">";
//voici le premier ajout : voici la table
if ($cellules<8 && $jour_semaine==date("w",
 mktime(0,0,0,$mois, 1, $annee))){
$jours=1;
}
//cette condition initialise la variable jours//////////////
if ($jours>0 && $jours<=date("t",$instant)){
```

Le format *t* correspond au nombre de jours dans le mois. Cette condition demande si le nombre `$jours` se situe entre 1 et le nombre de jours dans le mois. Si c'est le cas, on imprime le jour du mois.

```
if (date("j")==$jours && date("n")==$mois && date("Y")==$annee){
echo "<font color=\"#FFFFFF\" size=\"4\">";
}
```

```
else {
echo "<font color=\"#000000\" size=\"6\">";
}
echo "<div align=\"center\">$jours</div></font>";
}
else {
echo " ";
}
```

Le jour est affiché si la variable `jours` a été initialisée et si le nombre de jours dans le mois n'est pas dépassé. Si ce n'est pas le cas, on affiche le code HTML espace insécable (*Non Break Space*) pour que la bordure de la cellule soit affichée malgré tout :

```
echo "</TD>";
echo "</TR>";
///première ligne de la table puis deuxième ligne avec deux cases
echo "<TR height=\"50\">";
echo "<TD width=\"50\">";
if (in_array ($jours,$armoire_clefs)){
$clef=$rv[$jours][0];
$type=$rv[$jours][1];
$contenu=$rv[$jours][2];
echo "<A HREF=\"#\" onClick=\"ouvrir_vasistas('programmedujour.php?
clef=$clef','vasistas','menubar=no,scrollbars=no,statusbar=no,
 width=300,height=200')\">";
if ($type<1){
//ici nous affichons l'image qui correspond à un rendez-vous privé
 ou public
echo "<img src= \"priv_rv.gif\" width=10 height=10 border=0
 alt=\"$contenu\">";
}
else {
echo "<img src= \"pub_rv.gif\" width=10 height=10 border=0
 alt=\"$contenu\">";
}
echo "</A>";
}
echo "</TD>";
echo "<TD width=\"50\">";
/*ici vous pouvez mettre une condition pour afficher le saint du jour
echo $tous_les_saints[$jours]; mais nous préférons ajouter un lien
 pour ajouter un rendez-vous*/
echo "<img src= \"rendezvous.gif\" width=25 height=22 border=0
 alt=\"ajouter\">";
echo "</TD>";
echo "</TR>";
echo "</TABLE>";
```

```
echo "</TD>";
if ($jour_semaine==0){
echo "</TR>";
}
/////le dimanche finit la ligne/////////////
if ($jours>0){
$jours++;
}
if (date("t",$instant)==$jours && $jour_semaine==0){
break;
}
```

Avec le `break`, nous prévoyons le cas où le mois de février commence un lundi et comprend vingt-huit jours. Le break évite d'afficher une ligne de cases vides :

```
}
echo "<table>";
?>
</div>
```

LIMIT

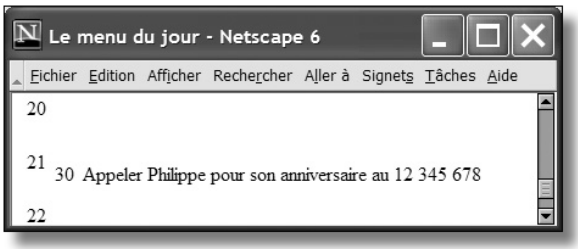
Vous pouvez limiter la requête SQL avec la syntaxe `LIMIT première_ligne [,dernière_ligne]`. Ainsi, le résultat ne contient pas les lignes de la requête commençant à `première_ligne`, et finissant à `dernière_ligne` ou à la fin des lignes appelées par défaut par la requête.

L'affichage du contenu du jour

Nous l'affichons dans une fenêtre avec la fonction Javascript `ouvrir_vasistas()` déjà étudiée. Le fichier se nomme :

```
$sql="SELECT datedujour, contenu FROM agenda WHERE clef='$clef'
 and nom_usage='$nom_usage'";
$resultat=@mysql_query($sql,$id_link);
while($rang=mysql_fetch_array($resultat)){
$datedujour=$rang['datedujour'];
$heure=preg_replace("/\d{8}(\d{2})(\d{2})\d{2}/","\\1",$datedujour);
$minutes=preg_replace("/\d{8}(\d{2})(\d{2})\d{2}/","\\2",$datedujour);
$contenu=$rang['contenu'];
$armoire[]=$heure;
```

```
/*nous rangeons toutes les clefs dans une armoire pour les déceler
 ensuite avec la fonction in_array()*/
$livredheures[$heure][]=$minutes;
$livredheures[$heure][]=$contenu;
/* nous créons un tableau multidimensionnel*/
}
```



▲ Figure 7.4 : *Le menu du jour*

Pour obtenir les minutes et les heures, nous utilisons une expression rationnelle qui pourrait être remplacée par un appel à la fonction `nombre_date()`. Les heures vont servir de clés car l'affichage se fait avec un index de 7 à 24 qui affiche les heures. Nous avons construit les coulisses ; définissons maintenant la scène :

Listing 7.4 : *Le menu du jour : lemenudujour.php*

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
    <title>Le menu du jour</title>
</head>
<body>
<table width="95%" cellspacing="2" cellpadding="2" align="left">
<?
for ($i=7;$i<24;$i++){
echo '<tr align="left" valign="middle">';
echo "\n<td>\n";
echo $i;
echo "<BR><BR>";
echo "\n</td><td>\n";
if (in_array($i,$armoire)){
$minutes=$livredheures[$i][0];
echo $minutes;
echo "\n</td><td>\n";
$contenu=$livredheures[$i][1];
```

```
echo $contenu;
echo "\n</td></tr>\n";
}
else{
echo "\n</td><td>\n";
echo "\n</td></tr>\n";
}
}
?>
</table>
</body>
</html>
```

L'index incrémente les heures, et les données sont extraites s'il apparaît que la clé est dans "l'armoire".

L'affichage de l'agenda sous forme de liste

Pour faire cet affichage de l'agenda sous forme de liste, il suffit de reprendre la requête de la page précédente et d'en changer la forme :

Listing 7.5 : *L'agenda sous forme de liste agenda_liste.php*

```
if (strlen($datedujour)==14){
$sql="delete from agenda where nom_usage='$nom_usage' and
 datedujour='$datedujour'";
$resultat=@mysql_query($sql,$id_link);
}
if (!$moischoisi){
$moischoisi=date("Ym");
}
$sql="select datedujour, contenu from agenda order by datedujour
 where nom_usage='$nom_usage' and datedujour LIKE '${moischoisi}%'";
$resultat=@mysql_query($sql,$id_link);
```

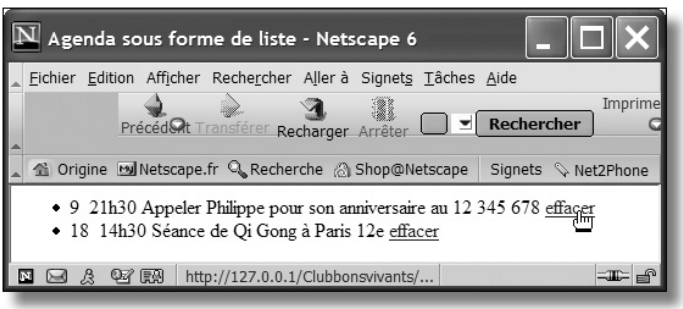
Une fonction Javascript demande confirmation au membre désireux d'effacer une entrée de l'agenda :

```
function confirmEffacer()
{
var accord=confirm("Vous voulez vraiment enlever cette entrée de
votre agenda ?");
if (accord)
    return true ;
else
    return false ;
}
```

Ici, il est disposé sous la forme d'une liste à puces :

Listing 7.6 : *l'agenda sous forme de liste* **agenda_liste.php**

```
echo "<UL>";
while ($rang= mysql_fetch_array($resultat)){
$contenu=$rang['contenu'];
if (strlen($contenu)>0){
$datedujour=$rang['datedujour'];
$heure=(int)substr($datedujour,10,2);
$minute =(int)substr($datedujour,12,2);

$lejour=(int)substr($datedujour, 6, 2);
$lejour.=" ${heure}h$minute";
echo "<LI>\t${lejour}\t${contenu}\t
<A HREF=\"$_SERVER['PHP_SELF']?moischoisi=$moischoisi&datedujour
=$datedujour\"
 onClick=\"return confirmEffacer()\">effacer</A>";
}
}
echo "</UL>";
```



▲ Figure 7.5 : *L'agenda sous forme de liste*

Dans la liste, vous accédez directement au contenu de l'agenda avec un lien pour effacer l'entrée.

Le formulaire pour écrire dans l'agenda

Pour entrer dans ce formulaire, nous ajouterons un menu dans la page calendrier de l'agenda. Ce menu nous donne le jour choisi sous la forme *AAAAMMJJ* et l'heure sous la forme *HHMM*.

Le formulaire est très simple, il contient un champ *textarea* nommé *contenu* et des boutons **radio** pour définir le type du message et son traitement. Pour modifier, ce sera aussi simple :

```
if ($_GET['cle']){
```

La variable s'appelle `cle` pour ne pas la confondre avec `clef`, donnée par l'authentification. Cette variable indique que le membre veut modifier son entrée dans l'agenda. Nous récupérons ensuite les données dans la table :

```
$cle=$_GET['cle'];
$sql="select * from agenda where clef='$cle' AND
 nom_usage='$nom_usage'";
$resultat=@mysql_query($sql,$id_link);
$rang=mysql_fetch_array($resultat);
$contenu=$rang['contenu'];
$datedujour=$rang['datedujour'];
$heure=substr($datedujour, 8, 2);
$minutes=substr($datedujour, 10, 2);
$type=$rang['type'];
}
?>
```

Maintenant, nous disposons le formulaire dans une table :

```
<table width="95%" border="0" cellspacing="10" cellpadding="10">
<tr>
      <td width="19%">  </td>
      <td width="81%">
<form name="form1" method="post" action="agenda.php">
    <textarea name="contenu" wrap="VIRTUAL" cols="80" rows="3">
<?
echo $contenu
?>
</textarea>
      </td>
    </tr><tr>
      <td width="19%">
<?
  if ($cle){
echo "<input type=\"hidden\" name=\"clefdesol\" value=\"$cle\">\n";
}
echo "<input type=\"hidden\" name=\"cettedate\"
value=\"$datedujour\">\n";
?>
```

Nous changeons le nom de la variable `datedujour` pour ne pas la confondre avec la même variable qui existe dans le fichier *agenda.php* où le formulaire est envoyé :

```
</td>
      <td width="81%">
        <select name="heure">
<option value=""></option>
<?
$heure=(int)$heure;
for ($i=7;$i<24;$i++){
if ($i==$heure){
$select=" SELECTED";
}
else {
$select='';
}
```

Ici, le bloc `else` permet de vider la variable. Si ce bloc n'existait pas, les derniers champs seraient toujours sélectionnés puisque la valeur ne changerait plus.

```
if ($i<10){
echo "<option value=\"0$i\"$select>$i</option>\n";
}
else {
echo "<option value=\"$i\"$select>$i</option>\n";
}
}
?>
        </select> heure
        <select name="minutes">
<option value=""></option>
<?
$minutes=(int)$minutes;
for ($i=0;$i<60;$i++){
if ($i%15>0){
continue;
//Nous n'affichons que les minutes des 1/4h
}
if ($i==$minutes){
$select=" SELECTED";
}
else {
$select='';
}
if ($i<10){
```

```
echo "<option value=\"0$i\"$select>0$i</option>\n";
}
//nécessité de faire précéder les unités d'un 0
else {
echo "<option value=\"$i\"$select>$i</option>\n";
}
}
?>
        </select> minutes
      </td>
    </tr>
    <tr>
      <td width="19%">  </td>
      <td width="81%">priv&eacute;
<?
 if ($type==1){
 $coche2=" CHECKED";
 }
 else {
 $coche1=" CHECKED";
 }
   echo "<input type=\"radio\" name=\"type\" value=\"0\"$coche1>\n";
   echo "public \n";
   echo "<input type=\"radio\" name=\"type\" value=\"1\"$coche2>";
?>
      </td>
    </tr><tr>
      <td width="19%"> </td>
      <td width="81%">
        <input type="submit" name="Submit" value="Envoyer"></form>
      </td>
    </tr>
  </table>
```

Enfin, nous ajoutons le traitement des données grâce au fichier *agenda.php*, en haut de l'application :

Listing 7.7 : *Formulaire pour l'agenda : agenda_ecrire.php*

```
if ($_POST['cettedate']){
$clefdesol =$_POST['clefdesol'];
$type =$_POST['type'];
$contenu =$_POST['contenu'];
$datedujour =$_POST['datedujour'];
$cettedate=substr($_POST['cettedate'], 0, 8);
$cettedate.=$heure.$minutes.'00';
if ($clefdesol==1){
$sql="UPDATE agenda SET type='$type', contenu='$contenu',
```

```
 datedujour='$cettedate' WHERE clef='$clefdesol' AND
nom_usage='$nom_usage'";
}
else {
$sql="insert into agenda (nom_usage, type, contenu , datedujour)
 VALUES('$nom_usage', '$type', '$contenu' , '$datedujour')";
}
@mysql_query($sql,$id_link);
//le nom d'usage sera donné par l'authentification
}
```

▲ Figure 7.6 : *Formulaire d'entrée pour l'agenda*

Voici donc un agenda pour les membres du club. Il sera intéressant de leur envoyer un courriel pour leur rappeler un rendez-vous ; mais là, c'est à vous de jouer !

7.3 Un miniforum sur chaque page

L'objectif est d'ouvrir certaines de vos pages à la discussion. Vous exposez, par exemple, un problème ou une expérience et désirez que les internautes réagissent. Rien de plus simple.

Commençons par récupérer le nom du fichier pour créer le nom de la table :

```
//////POUR RÉCUPÉRER LE NOM DU FICHIER SANS L'EXTENSION///
$extension=".php";
$table=basename ($_SERVER['PHP_SELF']);
////fonction basename()/////
$table=ereg_replace($table,$extension);
```

```
/*nous utilisons le nom du fichier sans son extension pour donner
 automatiquement un nom à la table*/
/*si votre version de PHP est au moins 4.1 vous pouvez mettre ces
 2 lignes en une seule*/
$table=basename ($_SERVER['PHP_SELF'],$extension);
Ici on récupère le nom du fichier et on en retire l'extension.
 Oui nous sommes toujours dans les commentaires!*/
//////////////FIN RÉCUPÉRATION////////////////////
```

Le problème ici est que certains caractères admis pour le nom d'un fichier sont interdits dans le nom d'une table *MySQL* :

```
//pour remplacer les quelques lignes au-dessus en anticipant un peu
preg_match("#^/.*/(.*)\.[A-z1-9]{3,4}$#", $_SERVER['PHP_SELF'], $tableau);
$table=$tableau[1];
///nous remplaçons les caractères interdits pour une table///
$caracteres_defendus=array('-', ' ');
while (list(, $valeur) = each ($caracteres_defendus)) {
$table=ereg_replace($valeur,'_',$table);
}
/*si le fichier a un nom uniquement composé de chiffres, il n'est pas
 possible de créer une table avec ce nom, nous ajoutons donc un
 espace
 blanc souligné. Vous aurez toutes les explications concernant les
 expressions rationnelles et en particulier ces quelques lignes dans
 le prochain chapitre. En fait, cette expression dit "si, dans le nom
 de la table, n'existe pas un caractère autre qu'un chiffre alors…".
 C'est une double négation*/
if (!ereg('[^0-9]',$table)){
$table.='_';
}
///////////////fin anticipation/////////////////////////////////
```

Nous utiliserons une expression rationnelle que nous étudierons dans le chapitre suivant, qui fonctionnera quelle que soit l'extension (*.php*, *.php3*, *.php4*, *.phtml*…). Si la table n'existe pas dans la base, nous la créons avec le nom forgé auparavant :

```
/*LIGNES DE CRÉATION DE LA TABLE QUE VOUS POURREZ ENLEVER OU PAS,
 UNE FOIS LA TABLE CRÉÉE*/
$sql1="SELECT clef from $table";
$resultat = @mysql_query($sql1, $id_link);
if (!$resultat){
$sql="CREATE TABLE $table (
clef int(11) NOT NULL auto_increment,
  date TIMESTAMP(12) NOT NULL default '0',
  nom varchar(60) NOT NULL default '',
  email varchar(60) NOT NULL default '',
```

```
  commentaire text NOT NULL,
  courriel char(1) NOT NULL default 'O',
  poste tinyint(4) NOT NULL default 'O',
  PRIMARY KEY  (clef),
  UNIQUE KEY clef_2 (clef),
  KEY clef (clef)
)";
@mysql_query($sql,$id_link);
//////////////////////FIN CRÉATION DE LA TABLE/////////////
```

Si le formulaire pour entrer un commentaire a été utilisé, la variable `commentaire` existe et il est temps d'insérer les données dans la table :

Listing 7.8 : *le haut du mini-forum* **mini-forum_haut.php**

```
$commentaire=$_POST['commentaire'];
if (isset($commentaire) && !empty($commentaire)){
$nouveau_commentaire=stripslashes($commentaire);
$sql="select email from $table where courriel='1' AND email !=''";
$resultat=mysql_query($sql,$id_link);
while ($rang=mysql_fetch_array($resultat)){
$nouvel_email=$rang['email'];
mail("$nouvel_email","Forum $table,"$nouveau_commentaire\n$nom\n",
 "From: $email");
}
   $commentaire=nl2br($commentaire);
   $date=time();
   $sql="insert into $table (date, nom, email, commentaire, courriel)
 VALUES ('$date', '$nom', '$email', '$commentaire', '$courriel')";
mysql_query($sql,$id_link);
}
?>
```

Avec cette application, nous avons rendu tout automatique pour que la table créée prenne le nom du fichier (attention de ne pas donner le même nom à plusieurs fichiers dans des répertoires différents). Vous pouvez laisser les lignes de création de la table car elles ne sont pas exécutées si la table existe avec au moins une ligne dedans. Cependant, elles ne servent plus à rien.

Affichons maintenant les commentaires déjà entrés, accompagnés du formulaire pour les saisir. Les champs n'ont pas besoin d'être vérifiés, seul le champ *commentaire* compte, et il est vérifié avant l'insertion des données dans la table. L'internaute peut cocher une case s'il veut recevoir les autres commentaires par courriel.

```
echo '<p align="center"><font face="Tahoma, Arial"><b>
VOS COMMENTAIRES</b></font></p>';
$sql = "SELECT * FROM $table";
$resultat=@mysql_query($sql,$id_link);
while ($rang=@mysql_fetch_array($resultat)){
$nom=$rang['nom'];
$email=$rang['email'];
$commentaire=$rang['commentaire'];
$date=$rang['date'];
$jourdhui=nombre_date($date);
//nous utilisons la fonction vue dans le chapitre 3
echo "<p><HR>auteur : <a href=\"mailto:$email?Subject=forum
 $table\">$nom</a>  <i>$jourdhui</i>.<br>";
echo "$commentaire";
}
```

Nous affichons maintenant le formulaire :

Listing 7.9 : *le bas du mini-forum* **mini-forum_bas.php**

```
echo "<p><hr><p>\n";
echo '<table align="center">';
echo "<tr><td>\n";
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
echo "Votre nom\n";
echo '</td><td><input type="text" name="nom" size="40"
maxlength="60">
</td></tr><tr><td>Votre email</td><td><input type="text"
name="email"
 size="40" maxlength="60"></td></tr><tr><td>Votre
 commentaire</td><td>
</td></tr><tr><td colspan="2"><textarea cols=80 rows=5
 name="commentaire"></textarea></td></tr><tr><td colspan="2">
Voulez-vous recevoir un email à chaque nouveau commentaire ?
 <input type="checkbox" name="courriel" value="1"></td></tr>
<tr><td><input type="submit" value="Insérez"></td><td></form></td>
</tr></table>';
echo "\n<p>\n";
```

7.4 Sondages et concours

Les sondages et les concours sont des formulaires comprenant des questions. Une fois le formulaire réalisé, il faut créer un affichage graphique pour le résultat du sondage.

Un sondage

Pour le sondage, créez une table dans la base pour les réponses, un fichier avec l'affichage des questions et l'affichage des résultats une fois les questions traitées. La table est très simple :

```
$sql="CREATE TABLE sonde (clef int(11) NOT NULL auto_increment,
 un int(11) NOT NULL default '0', deux int(11) NOT NULL default '0',
 trois int(11) NOT NULL default '0', quatre int(11)
 NOT NULL default '0', PRIMARY KEY (clef)) ";
@mysql_query($sql,$id_link);
```

Une fois la table créée, il faut insérer une ligne avec des 0 dans chaque colonne.

Cette table est destinée à accueillir plusieurs types de sondages ; voilà pourquoi le nom de ses colonnes est neutre. Cela évite de recréer à chaque fois une table. Pour un sondage, on considère que quatre réponses possibles sont suffisantes ; au-delà, le sondage devient une corvée pour les sondés.

Dans le programme de sondage, deux pages différentes sont superposées : la page du formulaire et la page du résultat montré graphiquement. L'aiguillage vers l'une ou l'autre de ces pages est actionné par trois variables : `$sondage`, `$un` et `$voir`. La première, `$sondage`, prouve que la personne a déjà rempli le formulaire en appuyant sur le bouton **submit**. La deuxième, que la personne a vu les résultats car la variable `$un` est initialisée uniquement dans cette partie. La troisième est une variable supplémentaire, utile pour installer un lien :
`<A HREF="sondage.php?voir=1">voir le résultat</A>`.

Si l'une des trois variables existe, c'est la page résultat qui doit être affichée sous forme graphique.

Variantes du programme en début de fichier

Dans ce programme, nous commençons par inclure les pages `.inc` nécessaires. Puis nous insérons, juste dessous, la liste des variables chargées de contenir les valeurs susceptibles de

changer d'un sondage à l'autre : `$table`, `$question`, `$reponse_1`, `$reponse_2`, `$reponse_3` et `$reponse_4`. Ainsi, vous pourrez copier le fichier en changeant simplement les valeurs de ces variables pour un nouveau sondage.

Le programme n'utilise qu'une seule ligne dans la table. Cette ligne, à chaque réponse, est actualisée dans la colonne correspondant à la réponse.

Création automatique de la table

Vous verrez dans le programme comment créer un programme en définissant automatiquement la table. Il suffit de faire une requête `SELECT`. Ensuite, le résultat est testé et, s'il est faux, la table reste à créer.

Pour mieux comprendre les résultats, chaque réponse affiche un graphique correspondant au pourcentage par rapport au nombre total des réponses. Pour faire ce graphique, il suffit de créer une image de 100 pixels de large et d'en afficher le pourcentage. Si la réponse représente 75 % des réponses, seuls 75 pixels de l'image sont affichés.

```
<?
include_once "../commun/fonctions.inc.php";
include_once RACINE."/commun/connexion.inc.php";
// APRÈS LES INCLUSIONS, LES VARIANTES SOUS FORME DE VARIABLES
$table='sondage';
$question="Quel vin préférez-vous ?";
$reponse_1="le vin rouge";
$reponse_2="le vin blanc";
$reponse_3="le vin rosé";
$reponse_4="le vin champagnisé";
$image_1="un.jpg";
$image_2="deux.jpg";
$image_3="trois.jpg";
$image_4="quatre.jpg";
////////////////////////////////////////////////////////////////
```

Cette requête SQL ne sert qu'à tester l'existence de la table et à la créer si elle n'existe pas :

```
$sql1="SELECT clef from $table";
$resultat = @mysql_query( $sql1, $id_link);
if (!$resultat){
$sql="CREATE TABLE $table (clef int(11) NOT NULL auto_increment, un
 int(11) NOT NULL default '0', deux int(11) NOT NULL default '0',
 trois int(11) NOT NULL default '0', quatre int(11) NOT NULL
 default '0',
 PRIMARY KEY (clef)) ";
@mysql_query($sql,$id_link);
$sql="insert into $table (un, deux, trois, quatre)
 values ('', '', '', '')";
///////////////// CRÉATION D'UNE LIGNE VIDE///////////////
@mysql_query( $sql, $id_link);
}
```

Nous abordons ici le traitement des données avec la mise à jour de la colonne correspondant au bouton coché dans le formulaire :

```
if (isset($sondage) || isset($un) || isset($voir)){
/////////ICI SELON LA RÉPONSE, LA COLONNE CHOISIE EST MISE À JOUR//
 if ($sondage=='un'){
$sql2="UPDATE $table SET un=un+1";
@mysql_ query($sql2, $id_link);
}
 elseif ($sondage=='deux'){
$sql2="UPDATE $table SET deux=deux+1";
@mysql_query($sql2, $id_link);
}
 elseif ($sondage=='trois'){
$sql2="UPDATE $table SET trois=trois+1";
@mysql_query($sql2, $id_link);
}
 elseif ($sondage=='quatre'){
$sql2="UPDATE $table SET quatre=quatre+1";
@mysql_query($sql2, $id_link);
}
$sql3="SELECT * from $table";
////EXTRACTION DU RÉSULTAT APRÈS LA DERNIÈRE MISE À JOUR///
$resultat = @mysql_query($sql3, $id_link);
$rang=mysql_fetch_array($resultat);
$un=$rang['un'];
$total=$un;
$deux=$rang['deux'];
$total+=$deux;
$trois=$rang['trois'];
```

```
$total+=$trois;
$quatre=$rang['quatre'];
$total+=$quatre;
```

Nous introduisons ici un test sur 0 pour éviter de générer une erreur :

```
if ($un>0){
$unpourcent=($un/$total)*100;
$unpourcent=(int)$unpourcent;
 }
```

Nous forçons le type pour obtenir un nombre entier. Nous aurions pu utiliser la fonction `floor()`, qui serait son équivalent. La fonction `floor()` renvoie l'arrondi inférieur avec `floor($unpourcent)` :

```
if ($deux>0){
$deuxpourcent=($deux/$total)*100;
$deuxpourcent=(int) $deuxpourcent;
}
if ($trois>0){
$troispourcent=($trois/$total)*100;
$troispourcent=ceil($troispourcent);
```

Pour que le total des pourcentages fasse 1, nous utilisons la fonction `ceil()`, qui renvoie l'arrondi supérieur :

```
}
if ($quatre>0){
$quatrepourcent=($quatre/$total)*100;
$quatrepourcent=ceil($quatrepourcent);
}
}
?>
```

Voyons maintenant le code HTML, qui est une imbrication entre le formulaire de participation au sondage et le résultat du sondage lui-même. L'affichage de l'un ou de l'autre se fait en fonction de l'existence de trois variables : `$sondage`, `$un` et `$voir`.

Listing 7.10 : *un sondage* **sondage.php**

```
<html>
<head>
<title>Le sondage des Bons Vivants</title>
<meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1">
```

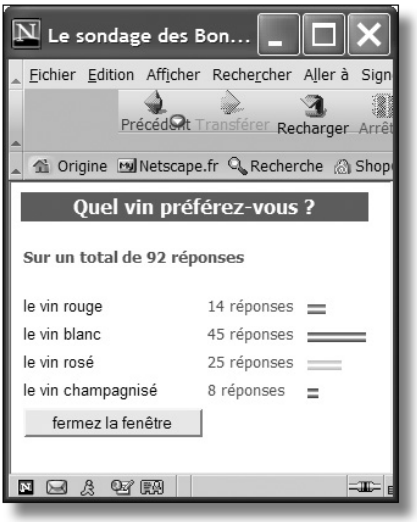
```
<script language="JavaScript">
<!--
function close_window() {
window.close();
}
//-->
</script>
</head>
<body bgcolor="#FFFFFF" text="#003366" link="#6699cc"
 vlink="#336699" alink="#FF0000">
///ICI COMMENCE LE TABLEAU//////////////////////
<table border="1" cellspacing="0" cellpadding="2" vspace="0"
 hspace="0" align="left">
  <tr align="center" valign="top">
    <td colspan="3" bgcolor="#336699">
<?
echo "<font face=\"Tahoma\" size=\"3\"><b><font color=\"#FFFFFF\">
$question</font></b></font>\n</td>\n</tr>";
echo "<form action=\"$_SERVER['PHP_SELF'] \" method=\"post\">";
if (isset($sondage) || isset($un) || isset($voir)){
echo '<tr align="left" valign="top">';
echo '<td colspan="3"> <br><font face="Tahoma" size="2"
 color="#1562AD"><b>';
echo "Sur un total de $total réponse";
if ($total>1){
echo 's';
//////////SI LE TOTAL DÉPASSE 1, LE MOT EST AU PLURIEL
}
echo "</b></font></td>";
echo '</tr>';
}
?>
//////////////////début première ligne et première réponse
  <tr align="center" valign="top">
    <td colspan="3" height="20"> <font face="Tahoma" size="2"
 color="#1562AD"><b></b></font>
    </td>
  </tr>
  <tr align="left" valign="top">
<?
echo "<td align=\"left\"><font face=\"Arial, Helvetica,
 sans-serif\" size=\"2\" color=\"#000000\">$reponse_1
</font></td>\n";
echo "<td> <font face=\"Tahoma\" size=\"2\" color=\"#1562AD\">\n";
if (!isset($un) && !isset($voir)){
////////SI C'EST LE FORMULAIRE, LE CHAMP EST AFFICHÉ//
echo '<input type="radio" name="sondage" value="un"></td>
<td> </td>';
}
```

```
else {
///////////////SINON C'EST LE NOMBRE DE RÉPONSES À 1///
echo "$un réponse";
if ($un>1){
echo 's';
}
echo "<td>  <img src=\"../images/$image_1\"
 width=\"$unpourcent\" height=\"8\" border=\"0\">";
}
?>
      </font></td>
  </tr>
/////////////FIN PREMIÈRE LIGNE//////////////////////////
/////////////DEUXIÈME LIGNE ET DEUXIÈME RÉPONSE/////////
  <tr align="left" valign="top">
<?
echo "<td align=\"left\"><font face=\"Arial, Helvetica, sans-serif\"
size=\"2\" color=\"#000000\">$reponse_2</font></td>\n";
echo "<td> <font face=\"Tahoma\" size=\"2\" color=\"#1562AD\">\n";
if (!isset($un) && !isset($voir)){
echo '<input type="radio" name="sondage" value="deux"></td>
<td> </td>';
}
else {
echo "$deux réponse";
if ($deux>1){
echo 's';
}
echo "</td><td>  <img src=\"../images/$image_2\"
 width=\"$deuxpourcent\" height=\"8\" border=\"0\">";
}
?>
      </font></td>
  </tr>
//////////fin deuxième ligne///////////////
/////////////troisième ligne et troisième réponse//////////
  <tr align="left" valign="top">
<?
echo "<td align=\"left\"><font face=\"Arial, Helvetica, sans-serif\"
size=\"2\" color=\"#000000\">$reponse_3</font></td>\n";
echo "<td> <font face=\"Tahoma\" size=\"2\" color=\"#1562AD\">\n";
if (!isset($un) && !isset($voir)){
echo '<input type="radio" name="sondage" value="trois"></td>
<td> </td>';
}
else {
echo "$trois réponse";
if ($trois>1){
echo 's';
}
```

```
echo "</td><td>  <img src=\"../images/$image_3\"
 width=\"$troispourcent\" height=\"8\" border=\"0\">";
}
?>
      </font></td>
  </tr>
//////////////////fin troisième ligne////////////////////
///quatrième ligne et quatrième réponse/////////////////
  <tr align="left" valign="top">
   <?
echo "<td align=\"left\"><font face=\"Arial, Helvetica, sans-serif\"
size=\"2\" color=\"#000000\">$reponse_4</font></td>\n";
echo "<td> <font face=\"Tahoma\" size=\"2\" color=\"#1562AD\">\n";
if (!isset($un) && !isset($voir)){
echo '<input type="radio" name="sondage" value="quatre"></td>
<td> </td>';
}
else {
echo "$quatre réponse";
if ($quatre>1){
echo 's';
}
echo "</td><td>  <img src=\"../images/$image_4\"
 width=\"$quatrepourcent\" height=\"8\" border=\"0\" >";
}
?>
      </font></td>
  </tr>
//////////////////fin quatrième ligne////
/////////ligne du bouton du formulaire ou de fermeture de la fenêtre/
  <tr align="left" valign="top">
    <td valign="top" align="left">
      <?
if (!isset($un) && !isset($voir)){
echo '<input type="submit" value="Envoyez">';
}
?></form>
      </td>
    <td align="right" colspan="2">  </td>
  </tr>
</table>
</body>
</html>
```

Au fur et à mesure, nous introduisons des méthodes qui faciliteront l'adaptation de vos programmes et vous feront gagner du temps. Vous pouvez utiliser n'importe quelle image, y compris employer la même image pour les quatre réponses. La méthode permet de se consacrer à

l'application plutôt que perdre du temps à copier du code ou à se replonger dans du code ancien dont vous ne connaissez plus les tenants ni les aboutissants.



◄ Figure 7.7 : *Sondage*

Pour appeler cette page dans une petite fenêtre, faites un lien avec un appel à la fonction Javascript nommée `ouvrir_vasistas()` que nous avons vue.

Un concours

Pour le concours, c'est le même système. Vous enlevez le graphique et ajoutez un champ pour le prénom, le nom et l'adresse e-mail ainsi que d'autres champs si vous le désirez. Le gagnant est tiré parmi les bonnes réponses. Le concours est clos un jour avant la proclamation des résultats. Vous allez donc créer une requête pour trouver le gagnant et le conserver sur deux fichiers avec des champs séparés.

Listing 7.11 : *Recherche du gagnant : recherche_concours.php*

```
$sql="select email, prenom, nom from concours where reponse2='1'
 ORDER BY RAND() LIMIT 1";
$resultat=@mysql_query($sql, $id_link);
```

```
$rang=mysql_fetch_array($resultat);
$email=$rang['email'];
$nom=$rang['nom'];
$prenom=$rang['prenom'];
$lenom="$prenom $nom";
$lenom =majuscules($lenom);
$contenu_txt="${lenom}§$email";
chdir('chemin/vers/le/répertoire/choisi/pourlesfichiers');
$fd_txt1=@fopen("resultat_concours1.dat",'w');
$fd_txt2=@fopen("resultat_concours2.dat",'w');
@fputs ($fd_txt1,$contenu_txt1, strlen ($contenu_txt1));
@fputs ($fd_txt2,$contenu_txt2, strlen ($contenu_txt2));
@fclose($fd_txt1);
@fclose($fd_txt2);
$message="Bravo $lenom,\nvous venez de gagner le
 concours.\nLe Webmaster";
mail ("$email","RESULTATS DU CONCOURS","$message","from:monsite.com");
```

Vérifiez que ce répertoire existe avec tous les droits d'écriture (`chmod()`). Mettez ce fichier dans votre répertoire protégé *administration*. Avant d'utiliser des bases de données, nous utilisions de simples fichiers. Ce système est utile ici car la requête est unique et le résultat, non reproductible. Il peut aussi être utile pour stocker les données et éviter, dans une page, de multiples requêtes SQL gourmandes en ressources serveur.

Voici le reste du programme que vous pouvez placer dans votre page d'accueil :

Listing 7.12 : *Le gagnant du concours : gagnant_concours.php*

```
$lejour=mktime (12, 0, 0, 9, 21, 2002);
$lejourdapres=mktime (0, 0, 0, 10, 1, 2002);
///le jour où vous arrêterez l'affichage
if (time()>$lejour && time()<$lejourdapres){
$fd_txt=@fopen('concours/resultat_concours1.dat','r');
$contenu_txt = @fread ($fd_txt,
 filesize ('concours/resultat_concours1.dat'));
@fclose($fd_txt);
if (!$contenu_txt){
$fd_txt=@fopen('concours/resultat_concours2.dat','r');
$contenu_txt = @fread ($fd_txt, filesize
 ('concours/resultat_concours2.dat'));
@fclose($fd_txt);
$fd_txt1=@fopen("resultat_concours1.dat",'w');
@fputs ($fd_txt1,$contenu_txt, strlen ($contenu_txt));
@fclose($fd_txt1);
```

```
///Nous réapprovisionnons le premier fichier//
}
$leschamps=explode('§',$contenu_txt);
$lenom=$leschamps[0];
echo "Le gagnant du concours est $lenom !";
}
```

Nous avons placé plusieurs champs dans le fichier pour que vous compreniez comment cela fonctionne. Vous pouvez ainsi mettre plusieurs lignes, chacune contenant des champs, et lire chaque ligne avec `fgets()` ou mettre le fichier dans un tableau avec une ligne par élément grâce à `file()`.

Les fonctions concernant le système de fichiers

Tab. 7.7 : Les fonctions pour lire et écrire un fichier

Fonction	Description
`int copy (string source, string destination)`	Copie le fichier.
`int delete (string fichier)`	Supprime le fichier.
`int fclose (int descripteur)`	Ferme le descripteur de fichier (voir `fopen`).
`int feof (int descripteur)`	Vrai si la fin du fichier est atteinte.
`string getc (int descripteur)`	Lit un caractère du fichier. Retourne faux si la fin du fichier est atteinte.
`string fgetcsv (int descripteur, int longueur [, string séparateur])`	Lit et retourne les champs d'un fichier *csv* dans un tableau (`;` comme séparateurs).
`string fgets (int descripteur, int longueur)`	Lit et retourne une ligne de fichier.
`string fgetss (int descripteur, int longueur)`	Lit et retourne une ligne de fichier qu'il débarrasse de ses balises.
`array file (string fichier)`	Lit un fichier qu'il retourne dans un tableau ligne par ligne. Si l'option est sur 1, le fichier sera recherché dans le chemin.
`int fopen (string fichier, string mode)`	Ouvre un fichier ou une URL et retourne un descripteur.

Tab. 7.7 : Les fonctions pour lire et écrire un fichier

Fonction	Description
`int fpassthru (int descripteur)`	Affiche le reste du fichier.
`int fputs (int descripteur, string car [, int longueur)`	Écrit dans un fichier. L'option longueur est recommandée.
`string fread (int descripteur, int longueur)`	Lit le fichier, au plus sur la longueur spécifiée.
`int fseek (int descripteur, int début)`	Place le pointeur sur une position donnée.
`mixed fscanf (int descripteur, string format [, string var1])`	Formate les données du fichier.
`int ftell (int descripteur)`	Retourne la position du pointeur interne.
`int fwrite (int descripteur, string car [, int longueur])`	Écrit une chaîne dans un fichier.
`int mkdir (string chemin, int mode)`	Crée un répertoire.
`int readfile (int fichier [, string chemin])`	Lit un fichier et en affiche le contenu.
`int rename (string vieux_nom, string nouveau_nom)`	Renomme un fichier.
`int rewind (int descripteur)`	Place le pointeur au début du fichier.
`int unlink (int fichier)`	Supprime un fichier.

Tab. 7.8 : Les modes d'ouverture de fichier

Mode	Description
`R`	Lecture seule.
`R+`	Lecture et écriture.
`W`	Écriture seule, écrase le fichier s'il existe.
`W+`	Lecture et écriture, écrase le fichier s'il existe.

Tab. 7.8 : Les modes d'ouverture de fichier

Mode	Description
A	Écriture seule, ajoute au contenu existant en plaçant le pointeur en fin de fichier.
A+	Lecture et écriture, ajoute au contenu en plaçant le pointeur en fin de fichier.

Tab. 7.9 : Informations sur le statut du fichier ou du répertoire

Fonction	Description
`int chmod (string fichier, int mode)`	Les permissions sont changées sur des répertoires ou des fichiers.
`int file_exists (string fichier)`	Vrai si le fichier existe.
`int fileatime (string fichier)`	Date du dernier accès au fichier.
`int filectime (string fichier)`	Date de modification du nœud.
`int filemtime (string fichier)`	Date de modification du fichier.
`int fileperms (string fichier)`	Droits du fichier.
`int filesize (string fichier)`	Taille du fichier.
`int filetype (string fichier)`	Type du fichier.
`Bool is_dir (string fichier)`	Vrai si c'est un répertoire.
`Bool is_executable (string fichier)`	Vrai si le fichier est exécutable.
`Bool is_file (string fichier)`	Vrai si c'est un fichier.
`Bool is_readable (string fichier)`	Vrai s'il est lisible.
`Bool is_writeable (string fichier)`	Vrai s'il est modifiable.
`Bool is_linkinfo (string fichier)`	Vrai si le fichier pointé existe.
`String readlink (string chemin)`	Nom du fichier pointé.
`array stat (string fichier)`	Information sur le statut.
`int touch (string fichier [, int time])`	Met à jour l'heure de modification.
`int umask (int masque)`	Définit un masque de permission par défaut (l'opposé de `chmod`).

chmod

Comme `chmod` utilise un nombre octal, il est indispensable de le noter précédé d'un 0, par exemple sous la forme 0755.

Voici les valeurs des permissions `chmod` pour Unix.

Tab. 7.10 : Permissions de fichier ou de répertoire de 0755

Permission	Propriétaire (owner)	Groupe (group)	Public (other)
Lecture (4)	x	x	x
Écriture (2)	x	o	o
Exécution (1)	x	x	x
Total	4+2+1=7	4+1=5	4+1=5

Ici, l'`umask` du fichier est 022, c'est-à-dire qu'il exclut la permission 2 (écriture) pour le groupe et le public. Vous affichez les interdictions plutôt que les permissions. Le `umask` sert de configuration par défaut sur un serveur pour une création de fichier. Il est configuré dans le répertoire *etc/profile/* avec Unix.

Les fichiers :

- *777*. Tout le monde peut lire, écrire et exécuter.
- *755*. Le propriétaire (vos programmes) peut tout faire, groupe et public peuvent lire et exécuter.
- *644*. Le propriétaire (vos fichiers) peut lire/écrire, groupe et public, seulement lire.

Les répertoires :

- *777*. Tout le monde peut lire, écrire et chercher.
- *755*. Le propriétaire peut tout faire, groupe et public peuvent seulement chercher.

Voici les réglages courants.

Les fichiers :

- scripts, *755* ;
- fichiers de données, *666* ;
- fichiers de configuration, *644* (pour les fichiers non modifiés par des programmes PHP).

Les répertoires :

- *777* (n'oubliez pas de mettre les permissions adéquates sur les fichiers à l'intérieur).

Tab. 7.11 : Arborescence

Fonction	Description
`bool chdir (string répertoire)`	Change le pointeur de répertoire courant.
`Void closedir (string descripteur)`	Ferme le descripteur de répertoire.
`int opendir (string chemin)`	Ouvre un répertoire.
`String readdir (int descripteur)`	Retourne le nom du fichier suivant associé au descripteur de répertoire.
`Void rewinddir (int descripteur)`	Place le pointeur au début du répertoire.
`int rmdir (string répertoire)`	Supprime un répertoire (vide).

Formater avec sprintf(), sscanf(), fscanf()

Tab. 7.12 : Le formatage des données

Signe	Source	Affichage
`%`	Pourcentage	Pourcentage littéral.
`b`	Entier	Binaire.
`c`	Entier	Caractère ASCII de cette valeur.
`d`	Entier	Nombre décimal signé.
`u`	Entier	Nombre décimal non signé.
`f`	Nombre réel	Nombre réel.

Tab. 7.12 : Le formatage des données

Signe	Source	Affichage
o	Entier	Nombre octal.
s	Chaîne de caractères	Chaîne de caractères.
x	Entier	Nombre hexadécimal dont les lettres sont des minuscules.
X	Entier	Nombre hexadécimal dont les lettres sont des majuscules.

7.5 Compteurs et statistiques

Les compteurs et les statistiques sont, comme le stéthoscope du médecin, des instruments d'écoute de l'activité cardiaque de votre site. Analyser la circulation sur votre site vous permet de voir quels sujets sont plébiscités, donc de projeter son développement futur.

La réflexion

Le compteur est une colonne mise à jour à chaque chargement de la page. Pour avoir des statistiques, il suffit que le compteur incrémente une ligne différente chaque jour. Ensuite, le jeu consiste à l'afficher en montrant les pages les plus souvent visitées et les statistiques quotidiennes et mensuelles. Vous pouvez complexifier à loisir, mais les statistiques horaires ou annuelles présentent un moindre intérêt.

Créer le compteur quotidien

Dans la table *compteur*, nous insérons les champs :

- *fois* pour le nombre de chargements de la page ;
- *date* sous forme *AAAAMMJJ* ;
- *page* pour insérer le nom de la page et placer ainsi toutes les pages sur la même table.

Créons la table :

```
CREATE TABLE compteur ( clef int(11) NOT NULL auto_increment,
 fois int(11) NOT NULL default '0', an char(2) NOT NULL default '0',
 mois char(2) NOT NULL default '0', jour char(2) NOT NULL default '0',
 repertoire varchar(255) NOT NULL default '',page varchar(60)
 NOT NULL default '', PRIMARY KEY (clef), KEY clef (clef))
```

Le principe est simple. Si la date est différente de la dernière date, on crée une nouvelle ligne dans la table ; autrement, on se contente de faire une mise à jour du champ *fois* qu'on incrémente.

La page est identifiée par la variable globale, `$SCRIPT_FILENAME`, qui donne le nom du fichier et du répertoire depuis la racine du site (au-dessus du répertoire *www* ou *htdocs*) :

Listing 7.13 : *Les statistiques* **stats.inc.php**

```
<?
$table='compteurs';
$chemin=ereg_replace('/le chemin vers/www','',$SCRIPT_FILENAME);
$repertoire=dirname($chemin);
$page=basename($chemin);
$sql= "select * from $table where page='$page'
 AND repertoire='$repertoire' ORDER BY clef DESC LIMIT 1";
//vérifions la dernière date entrée dans la table pour cette page
$resultat=@mysql_query($sql,$id_link);
$rang=@mysql_fetch_array($resultat);
$jour=$rang['jour'];
$mois=$rang['mois'];
$an=$rang['an'];
if ($jour!=date("d")){
//si la date est différente, la nouvelle ligne est créée
$an=date("y");
$mois=date("m");
$jour=date("d");
$sql= "insert into $table (fois, an, mois, jour, repertoire, page)
 VALUES('1', '$an', '$mois', '$jour', '$repertoire','$page')";
}
else {
//sinon elle est juste mise à jour
$sql= "update $table set fois=fois+1 where jour='$jour' AND mois=
'$mois' AND an='$an' AND page='$page' AND repertoire='$repertoire'";
}
@mysql_query($sql,$id_link);
?>
```

Pour appeler maintenant ce programme, il vous suffit de le nommer `stats.inc.php` et de l'inclure dans vos pages par la ligne :

```
include_once RACINE."/commun/stats.inc.php";
```

$SCRIPT_NAME ou $SCRIPT_FILENAME

Vous pourriez utiliser `$SCRIPT_NAME`, cela vous éviterait d'enlever le chemin entre la racine et votre répertoire *www*. Cependant, si vous avez des sous-domaines comme `http://vins.bonsvivants.com`, `$SCRIPTNAME` vous donnera le même répertoire que pour `http://www.bonsvivants.com`, c'est-à-dire *"/"*. Pour éviter ce problème, il vaut mieux utiliser `$SCRIPT_FILENAME` et en enlever le chemin au-dessus du répertoire web.

Tab. 7.13 : Découpage du chemin d'un fichier

Fonction	Description
`String basename (string chemin)`	Extrait le nom du fichier d'un chemin.
`String dirname (string chemin)`	Retourne le chemin sans le nom de fichier.

La page d'affichage des statistiques de page

Si vous avez bien conçu votre site, vous avez groupé vos applications par répertoire. Il devient alors intéressant de disposer de statistiques par répertoire en plus des statistiques par fichier. Quant aux statistiques quotidiennes et mensuelles, sans doute disposez-vous de chiffres fournis par votre hébergeur. Les vôtres les compléteront et les affineront. Par ailleurs, vous avez un certain nombre de fichiers chargés à chaque page. Ce n'est pas la peine d'inclure ces derniers dans vos statistiques.

Pour la page, nous utiliserons des fonctions de manipulations de fichiers et de répertoires.

La page comprend la liste des répertoires avec leurs statistiques par mois classées par ordre décroissant. Pour chaque répertoire affiché, un menu permet de choisir un fichier dont les statistiques s'ouvriront dans une nouvelle fenêtre.

Listing 7.14 : *Affichage des statistiques : affichage_stats.php*

```
<?
$extensions=array ('htm', 'html', 'php', 'php3');
echo '<TABLE width="95%">';
$sql="select SUM(fois) as somme, repertoire, an, mois from compteurs
 GROUP BY repertoire, an, mois ORDER BY an, mois, somme DESC";
$resultat=@mysql_query($sql,$id_link);
while($rang=@mysql_fetch_array($resultat)){
$somme=$rang['somme'];
$repertoire=$rang['repertoire'];
$an=$rang['an'];
$mois=$rang['mois'];
echo "<TR><TD width=\"18%\">";
echo "$repertoire</TD>";
$dernier_repertoire=$repertoire;
echo "</td><TD width=\"20%\">\n";
echo "<form action=\"stats_fichier.php\" method=\"post\"
 target=\"_blank\">\n";
echo "<select name=\"fichier\">\n";
$sql_fichiers="select page DISTINCT from compteurs where
 repertoire='$repertoire' ORDER BY page";
$result=@mysql_query($sql_fichiers,$id_link);
while($rangee=@mysql_fetch_array($result)){

$page=$rangee['page'];
preg_match("#^(.*)\.([A-z0-9]{3,4})$#", $page, $tableau);
$extension=$tableau[2];
/* NOUS TRICHONS UN PEU style "Retour vers le futur". Nous vous
 expliquerons la manipulation dans le prochain chapitre. Promis! */
in_array ($extension, $extensions)){
echo "<option value=\"";
echo "$repertoire$page";
echo "\">$page</option>\n";
}
}
echo "</select>\n";
echo "<input type=\"submit\" value=\"voir\">\n";
echo "</form>";
echo "</TD><TD>";
if ($an!=$dernier_an){
echo $an;
$dernier_an=$an;
}
echo "</TD><TD>";
echo $mois;
echo "</TD><TD>$somme</TD></TR>";
}
echo"</TABLE>";
?>
```

Par le formulaire et son menu déroulant, nous envoyons le chemin du fichier dont nous désirons afficher les statistiques. C'est le même fichier. Dans le tableau HTML, il vous suffit d'enlever la colonne du formulaire et d'ajouter une colonne pour afficher le jour.

N'oubliez pas d'ajouter les deux lignes suivantes :

```
$repertoire=dirname($fichier);
$page=basename($fichier);
```

Ensuite, la requête SQL est :

```
$sql="select SUM(fois) as somme, fichier, an, mois, jour from
 compteurs GROUP BY fichier, an, mois, jour ORDER BY an,
 mois, jour somme DESC";
```

7.6 Le chariot électronique

Imaginons que, dans ce club, un marchand de vin ou un producteur vous propose de mettre une boutique en ligne.

Une boutique est avant tout une calculatrice qui suit le client de page en page durant sa visite. Nous gardons les informations pendant la visite. Elles correspondent aux articles achetés avec leur clé dans la table et leur quantité.

Le catalogue consiste en une liste de produits dans chaque page, accompagnés d'une photo, d'un champ *quantité* et d'un bouton pour acheter. Cliquer sur le bouton ajoute dans le tableau en session une ligne comprenant la clé du produit et sa quantité. Une fenêtre s'ouvre et confirme que le produit a été ajouté dans le chariot. Dans chaque page sont affichés deux boutons, l'un pour voir le contenu du chariot et l'autre pour finir la visite et payer les marchandises avec le port. Ce tableau est construit avec comme clé la référence du produit (la clé dans la table *produits*) et comme valeur, la quantité.

Le premier travail consiste à concevoir et à construire la base de données. Dans l'exemple d'un vin, nous pouvons affiner les caractéristiques : nature (blanc, champagne, rosé, rouge, mousseux), cépage, terroir, pays, année, appellation, commentaire, contenance, prix, promotion. Ensuite,

vous créez une table avec ces caractéristiques. Nous pouvons intégrer dans l'identifiant `référence` un certain nombre de renseignements qui économiseront de la place et du temps de frappe. Nous commençons par les trois premiers caractères du type puis une lettre tirée d'une table caractérisant le cépage… Dans notre exemple, nous utilisons le vin, mais vous pouvez faire la même chose avec des livres, de l'épicerie, des séjours à la montagne… Seules les caractéristiques changeront. Avant de commencer, visitez plusieurs sites sur le même thème.

La promotion

La valeur `promotion` peut être constituée de la date de début, de la date de fin, de la quantité et du prix sous la forme `2002090220020917330`. Dans l'exemple, la promotion commence le 2 septembre 2002 (8 premiers chiffres 2002 09 02) et finit le 17 septembre 2002 (du 9e au 17e chiffre) ; enfin, l'offre consiste en trois bouteilles (18e chiffre) pour 30 €, ou 30 $ (19e et 20e chiffres). Voilà des astuces qui économisent de la place dans la base.

Vous créez une fonction qui décode la référence et la promotion. Ce livre n'étant pas spécialisé dans le commerce des vins, nous n'entrerons pas dans le détail des références de vins. Si vous voulez approfondir, visitez l'excellent site `www.vitis.org`. Étudions de plus près la fonction `promotion()` :

Listing 7.15 : *Transformation du chiffre de la colonne promotion : fonction_promotion.php*

```
$mois_franc=array('', 'janvier', 'février', 'mars', 'avril', 'mai',
 'juin', 'juillet', 'août', 'septembre', 'octobre', 'novembre',
 'décembre');
echo promotion("2002091020020929330")
function promotion($chiffre){
$debut=substr($chiffre,0,8);
$an=substr($debut,0,4);
$mois=(int)substr($debut,4,2);
$mois= $mois_franc["$mois"];
$jour= (int) substr($debut,6,2);
$debut="$jour $mois $an";
$promo['debut']=$debut;
$fin=substr($chiffre,8,8);
$an=substr($fin,0,4);
$mois=(int)substr($fin,4,2);
$mois= $mois_franc["$mois"];
$jour= (int) substr($fin,6,2);
```

```
$fin="$jour $mois $an";
$promo['fin']=$fin;
$nombre=substr($chiffre,16,1);
$promo['nombre']=$nombre;
$prix=substr($prix,17,2);
$promo['prix']=$prix;
return $promo;
}
```

Le prix affiché doit être formaté, car les nombres à virgule flottante sont écrits avec un point pour les décimales alors qu'en français on utilise une virgule. Si un nombre comprend une virgule, il devient une chaîne de caractères. Il existe une fonction pour remédier à ce problème.

Les prix

Pour afficher vos prix, vous devrez les formater pour le français. Dans la table *produits*, ce sont des nombres à virgule flottante où la virgule est représentée par un point.

number_format()

Cette fonction sert à formater un nombre pour le conformer aux usages locaux.

Syntaxe : `string number_format ( float nombre [, int decimals [, string point_dec [, string sep_milliers]]])`

Pour formater un nombre en français, le code serait :

```
$prix_total=number_format ($prix_total, '2', ',', ' ');
```

Dans cet exemple, nous afficherons un nombre qui comprend deux décimales après une virgule et utilise une espace pour séparer les milliers.

Le nombre formaté

Le nombre formaté a changé de nature. Il est non plus un nombre à virgule flottante mais une chaîne de caractères. Cela signifie que, si vous le traitez avec un opérateur arithmétique, il devient un entier et perd ses décimales.

Les tables à créer

Dans la table *transport*, vous avez deux champs, *port* et *prix*. Le chiffre inscrit dans la colonne `prix` correspondra à la limite supérieure de l'intervalle de prix pour un port donné. La table *clients* dépendra des champs du formulaire. Inspirez-vous de celui que nous avons vu au chapitre *MySQL en action*. La table *commandes* est un duplicata de la table *produits* à laquelle vous ajoutez une colonne pour insérer la clé de la table *clients*. Enfin, la table *paiements* contiendra la `clef` du client, la somme totale, la date et l'heure ainsi que le mode de paiement. Vous pouvez ajouter une table *modes_paiement* si vous le désirez, avec un code et un nom pour chaque mode.

L'action "acheter un produit"

Dans votre catalogue, vous affichez vos produits avec la photo, les caractéristiques. Quand l'internaute clique sur le bouton dans l'intention d'acheter le produit, il ne fait qu'envoyer un formulaire. Ce formulaire peut être envoyé à la page elle-même ou à une autre page qui s'affiche dans une fenêtre. Dans les deux cas, vous confirmez l'achat du produit avec toutes les précisions.

Pour l'affichage d'un certain nombre de données, voyez l'affichage d'un carnet de liens dans le chapitre *Moteurs de recherche et expressions rationnelles*.

Le code du formulaire serait celui-ci :

```
<form action="ajouter_chariot.php" method="post" target="chariot">
<?
echo "<input type=\"hidden\" name=\"clef\" value=\"$clef\">";
?>
<input type="text" name="quantite" value="1" size="2">
<input type="submit" value="j'achète">
</form>
```

Une seule table de produits

Ce raisonnement part du principe que vous avez une seule table pour tous vos produits ou, pour le moins, une clé unique pour chaque ligne parmi tous les produits.

Maintenant, pour traiter les données et afficher la confirmation, votre code ressemblera à celui-ci :

```
<?
session_start();
$chariot["$clef"]=$quantite;
session_register("chariot");
include "../commun/connexion.inc.php";
include "../commun/fonctions.inc.php";
include "../commun/stats.inc.php";
```

Vous affichez ensuite le contenu du chariot.

L'action "voir le contenu du chariot"

L'action "voir le contenu du chariot" consiste à afficher le contenu de la table *produits* en regard des numéros de clés, accompagné de la quantité et du prix. Pour voir le contenu, faites sur le tableau une boucle `foreach` dans laquelle vous envoyez une requête SQL. Sorti de la boucle, multipliez la quantité par le prix à l'unité pour obtenir le prix correspondant à chaque article :

```
<TABLE width="95%">
<TR><TD>….</TD><TD>unité</TD><TD>qté</TD><TD>total article</TD></TR>
<?
if ($action =='voir'){
foreach ($chariot as $cle=>$val){
echo "$cle=>$val<br>";
$sql="select * from produits where clef='$cle'";
$resultat=@mysql_query($sql,$id_link);
$rang=mysql_fetch_array($resultat);
//PRENONS PRIX PAR EXEMPLE//////
$prix_unit=$rang['prix'];
$prix_total_article=$prix.*$val;
$sous_total_prix+=$prix_total_article;
//NOUS CALCULONS LE SOUS-TOTAL////
```

```
$prix_total_article =number_format($prix_total_article,'2',',',' ');
$prix_unit =number_format ($prix_unit, '2', ',', ' ');
echo "<TR><TD>….</TD><TD> $prix_unit</TD><TD>
<form action="ajouter_chariot.php" method="post" target="chariot">
<?
echo "<input type=\"hidden\" name=\"clef\" value=\"$clef\">";
echo "<input type=\"text\" name=\"quantite\" value=\"1\" size=\"2\">
</TD>\n";
echo "<TD>$prix_total_article</TD>\n<TD>
<input type=\"submit\" value=\"je change la quantité\">
</form></TD>\n";
echo "<TD><form action=\"ajouter_chariot.php\"
 method=\"post\" target=\"chariot\">\n";
echo "<input type=\"hidden\" name=\"act\" value=\"f\">";
echo "<input type=\"hidden\" name=\"clef\" value=\"$clef\">";
?>
<input type="submit" value="enlever"></form </TD></TR>
}
echo "<TR><TD>….</TD><TD> </TD><TD>total article</TD>
<TD>$sous_total_prix </TD></TR>\n";
```

Au bas de l'affichage du chariot, vous afficherez le port pour l'envoi, le total et un bouton **commander**. Ensuite, vous enregistrerez les coordonnées du client et placerez tout le contenu du chariot dans une table :

```
$sql="select port from transport where prix>$sous_total_prix
 ORDER BY prix LIMIT 1";
$resultat=@mysql_query($sql,$id_link);
$rang=mysql_fetch_array($resultat);
$port=$rang['port'];
$prix_total=$sous_total_prix+$port;
$prix_total=number_format ($prix_total, '2', ',', ' ');
echo "<TR><TD>….</TD><TD> </TD><TD>port</TD><TD>$port </TD></TR>\n";
echo "<TR><TD>….</TD><TD></TD><TD>Total</TD>
<TD>$prix_total </TD></TR>\n";
}
```

Vous aurez remarqué que `$sous_total_prix` n'est pas encadré de guillemets simples car il s'agit d'une quantité.

L'action "commander les articles"

Pour créer la commande, il vous reste à définir un formulaire avec des champs cachés dont les valeurs sont fournies par les variables de chaque article. Le bouton **commande** envoie les données dans la table *commandes* et ouvre le formulaire d'enregistrement du client. Dans ce

formulaire, nous conservons la clé de la table *commandes* avec la fonction `mysql_insert_id()`, puis l'insérons dans un champ caché pour avoir la possibilité d'ajouter la clé de la table *clients*, une fois les coordonnées de l'internaute enregistrées. Pour ajouter ce numéro dans la table *commandes*, insérez ces lignes juste après la commande SQL pour la table *clients* :

```
$sql="insert into commandes (les différents champs)
 VALUES (les différentes valeurs)";
@mysql_query($sql,$id_link);
$clef_commande=mysql_insert_id();
```

puis dans le formulaire :

```
<input type="hidden" name="clef_commande" value="$clef_commande">
```

Une fois les coordonnées du client insérées dans la table *clients*, ajoutez dans la table *commandes* :

```
$clef_client=mysql_insert_id();
$sql="update commandes set $clef_client='$clef_client'
 where clef='$clef_commande'";
@mysql_query($sql,$id_link);
```

mysql_insert_id()

Cette fonction correspond à la fonction `MySQL : last_insert_id()`. Elle permet donc de récupérer la dernière valeur d'une colonne AUTO_INCREMENT.

En haut du fichier, pour traiter la suppression d'un article par l'internaute qui aurait cliqué sur le lien *enlever*, nous ajoutons le code suivant :

```
if ($_GET['act']=='f'){
$autre_chariot["$clef"]=$quantite;
$chariot=array_diff($chariot, $autre_chariot);
}
```

La fonction `array_diff()` recrée le tableau `$chariot` en enlevant les éléments communs avec le tableau `$autre_chariot`. Ainsi, l'article en question sera effacé.

Fonctions d'arrondi augmentées des fonctions maximum et minimum

Tab. 7.14 : Fonctions d'arrondi augmentées des fonctions maximum et minimum

Fonction	Description
`mixed abs (mixed nombre)`	Valeur absolue (non signée) du nombre.
`int ceil (float nombre)`	Arrondi à l'entier supérieur.
`int floor (float nombre)`	Arrondi à l'entier inférieur.
`mixed max (mixed nombre1, mixed nombre2…)`	Plus grande valeur de la liste.
`mixed min (mixed nombre1, mixed nombre2…)`	Plus petite valeur de la liste.
`float round (float nombre [, int precision])`	Arrondi. La précision optionnelle donne le nombre de décimales après la virgule.

7.7 Un magazine

Le magazine utilise des `templates`, qui sont des modèles, des gabarits ou des moules dans lesquels nous coulons du texte. Nous employons plusieurs techniques pour cela. Le modèle est défini sur un logiciel de création de pages web. Nous utilisons du faux texte, extrait de n'importe quel texte numérique. Les professionnels utilisent des textes en latin. Une fois le modèle achevé, nous remplaçons les différents textes par des balises. Puis le programme créera les fichiers fils à partir du modèle, les fichiers qui, eux, seront véritablement affichés.

Le processus de création d'un numéro se fait à partir d'un formulaire qui envoie du texte dans une table. Les colonnes ont le même nom que les balises (numéro, rubrique, page, titre, accroche, chapô, article…). Une fois le texte entré dans la base, le programme récupère le texte du modèle et remplace, dans le fichier, les balises par le contenu des champs de même nom avec une expression rationnelle. Enfin, il crée un fichier dans lequel il coule le texte contenu.

Si vous voulez modifier l'article, il vous suffit ensuite de changer les données dans la base. Vous écraserez ainsi le fichier précédent, qui possède un nom identique.

Pour modifier le design de vos pages, travaillez sur le modèle. Le contenu et le design sont clairement séparés, ce qui donne plus de souplesse. Vous travaillez sur un modèle et non sur chaque page. Ensuite, le changement se fait en un clic.

La réflexion

Cette application demande une longue réflexion. La maquette doit être faite comme pour un magazine papier. Chaque page et article sont calibrés avec le nombre de signes, la police…

Pour l'abonnement, nous allons utiliser la table *membres*, dans laquelle nous ajouterons une colonne `abonnement` avec la date du dernier mois d'abonnement (`AAAAMM`), puisque cette revue est mensuelle.

Le formulaire d'abonnement est très simple, sans code PHP. La personne accède à la page par son identité de membre. Un formulaire contient un menu déroulant. Ce dernier donne la périodicité (trois, six, douze ou vingt-quatre mois).

La première page du magazine affiche le sommaire et ne demande pas d'authentification ; en revanche, le magazine lui-même demande l'authentification et la validité de la date d'abonnement.

Les paramètres pour un article pourraient être la clé, le numéro, la rubrique, la page, le titre, l'accroche, le chapô, l'article, l'auteur et la date de la parution.

Le sommaire cherchera le titre, l'accroche et l'auteur pour chaque article dans la table, en fonction du numéro en cours. Les articles, en revanche, seront des pages statiques créées "en dur« par l'application, une fois pour toutes, à partir de la base de données.

Dans l'administration, une page affiche les abonnements en cours ou périmés avec un bouton **prolonger**. En effet, il se peut que vous ne disposiez pas du paiement électronique et receviez vos paiements par chèque.

Un formulaire permet d'entrer les articles dans la table. Une date de parution est décidée. Elle sera ajustée avec un formulaire capable de modifier la date de parution de ce numéro. Dans le répertoire d'administration, le magazine à paraître sera visible pour constater le travail effectué et ce qui reste à modifier. Plusieurs magazines seront en chantier avant la date de parution, jusqu'à ce qu'ils soient achevés.

Les images possèdent un nom qui comprend la place dans la page, le numéro de page et le numéro du magazine. La page peut être divisée en neuf et codée (*hg* pour haut gauche, *md* pour milieu droit, *mm* pour milieu milieu, c'est-à-dire centre…). La première image située dans le milieu gauche de la page 2 du numéro 4 s'appellera *012mg4*, la deuxième image située dans le haut gauche, *022hg4*, etc.

Une fois la date arrivée, le numéro sera visible aux abonnés et un courriel, envoyé automatiquement pour l'annoncer. Une application pour envoyer les courriels sera créée. Cette application enverra un courriel trois mois, puis un mois avant l'échéance de l'abonnement, pour demander à l'abonné s'il veut prolonger son abonnement. Quelques jours avant la parution du nouveau numéro, il recevra un sommaire du numéro à paraître. S'il ne se réabonne pas, il recevra un questionnaire pour connaître les raisons de son désabonnement. Dans cette application, la plus grosse partie de votre temps sera consacrée à la réflexion et à l'étude du projet. La partie la plus lourde du code utilisé est déjà contenue dans ce livre. Nous allons diriger le projecteur sur des aspects particuliers de l'application, comme étudier le procédé pour télécharger les images sur le serveur par un formulaire.

Le formulaire pour entrer les images

Les images sont regroupées dans un répertoire correspondant au numéro du magazine, créé à la mise en chantier d'un numéro. Elles sont répertoriées dans une table *images_mag*. À chaque image chargée, le nom est décodé et la colonne `nbre_images`, incrémentée de 1 sur la ligne correspondant à l'article et au numéro. Voici le formulaire dans le fichier nommé *images_mag.php* :

```
<?
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"POST\"
 enctype=\"multipart/form-data\">";
```

```
?>
<table border="0" width="80%"><tr><td>
<input type="Hidden" name="MAX_FILE_SIZE" value="2097152">
Fichier &agrave; transf&eacute;rer : </td><td>
<input type=file name="userfile">
</td></tr><tr><td></td><td align="center">';
<input type="Submit" value="envoyer">
<td></tr></table>
</form>
```

Télécharger une image

Avant de créer un formulaire pour télécharger une image, vérifiez avec la fonction `phpinfo()` les paramètres `upload_max_filesize` et `upload_tmp_dir`, qui renseignent respectivement sur la taille maximale des fichiers téléchargés et le répertoire qui sert de sas aux images avant qu'elles ne rejoignent leur répertoire de destination.

▲ Figure 7.8 : *Formulaire de téléchargement*

Ajoutons maintenant le traitement de formulaire en haut du fichier. Trois caractéristiques : `enctype="multipart/form-data"` dans la balise `form` , `MAX_FILE_SIZE` en champ caché et `<input type=file name="userfile">`, qui est un champ de chargement de fichiers qui affichera automatiquement un bouton **parcourir** pour que l'internaute puisse aller chercher le fichier sur son disque dur.

Listing 7.16 : *le formulaire de téléchargement de l'image images_mag.php*

```
<?
if ($MAX_FILE_SIZE==2097152){
if ($userfile_size<$MAX_FILE_SIZE){
copy($userfile, "F:/Program Files/EasyPHP/tmp/"."$userfile_name");
$car_image=GetImageSize("$userfile");
$largeur=$car_image[0];
$hauteur=$car_image[1];
if (($largeur+$hauteur>500) || $largeur>300 || $hauteur>300){
    echo '<html><head><title>Erreur</title></head><body>';
    echo 'Votre image est trop grande, vous ne pouvez dépasser
 les 300 x 200 pixels ou 200 x 300 pixels.<br>';
echo "Revenir au <a href=\"images_mag.php\">formulaire</a>.";
    echo '</body></html>';
 exit();
}
$taille=$car_image[3];
$extension=substr($userfile_name,-4);
/* Le signe - signifie que nous commençons à la fin de la chaîne de
caractères. */
$nom_image.=$extension;
  $sql="UPDATE magazine SET image_$colonne='$nom_image',
 legende_$colonne='$legende', taille_$colonne='$taille'
 where nom_fichier='$nom_fichier'";
mysql_query($sql, $id_link);
copy("F:/Program Files/EasyPHP/tmp/$userfile_name",
 "F:/Program Files/EasyPHP/www/images_magazine/$nom_image");
chmod("F:/Program Files/EasyPHP/www/images_magazine/$nom_image",0766);
unlink($userfile);
}
else {
$annonce="<p><font color=\"#FF0000\">Votre fichier est trop gros,
 son poids ne peut dépasser 2Mo</font></p>";
}
}
?>
```

Vous donner une application complète de ce type demanderait un livre entier. Le raisonnement à la base de l'envoi d'images comme de l'envoi d'articles est le suivant : la table pour chaque numéro est créée avec tous les noms d'articles et tous les noms d'images. Les champs qui les accompagnent comme `legende`, `taille` pour l'image, `titre`, `accroche`, `chapo`… sont vides. À chaque entrée, les champs sont complétés pour le nom ou la rubrique qui correspond. Cela permet de mettre des menus déroulants dans les formulaires d'entrée afin de choisir le numéro de magazine, le numéro de page… et ainsi de composer un nom qui existe

déjà dans la table. Il suffit alors de compléter la ligne correspondant à ce nom, après l'envoi du formulaire.

Vous l'aurez compris, la difficulté réside non pas dans le code mais dans l'architecture de l'application. Dans le prochain chapitre, nous verrons comment remplacer les balises du magazine par le contenu de la table dans les fichiers et comment élaborer un moteur de recherche dans les numéros et les pages du magazine.

7.8 Une galerie photo

L'application consiste à afficher des miniatures de photos dans plusieurs pages reliées. Vous pourrez utiliser un formulaire du type de celui étudié ci-dessus pour entrer les images. Celles-ci sont mises dans un répertoire qui sert de sas, en attendant d'être traitées. Ce formulaire permet d'entrer les informations concernant l'image dans la base de données. Le programme crée une copie miniature avec un préfixe. Sans entrer dans le détail de la pagination, nous allons voir le formulaire et son traitement et l'affichage des pages.

Pour avoir plus de détails sur la pagination, reportez-vous à l'annuaire de liens dans le chapitre *Moteurs de recherche et expressions rationnelles* ou au magasin électronique dans le chapitre *Parsez-vous XML ?*

Le formulaire est un formulaire de téléchargement que nous avons déjà vu dans ce chapitre. À part les champs texte que vous y ajoutez, nous allons voir comment mettre un champ menu qui affiche les différentes galeries en lisant le répertoire principal et y relevant les différents dossiers qui y sont contenus :

```
<select name="galeries" size="1">
<?
$dossier='galeries';
$repertoire=openDir(galeries);
while ($galerie=readDir($repertoire)){
if (($galerie !=".") && ($galerie !="..") && (!is_file($galerie))){
echo "<option value=\"$galerie\">$galerie</option>";
}
}
?>
</select>
```

Voici la fonction qui crée une miniature de l'image proportionnelle à la taille de l'image, ce qui permet d'avoir une certaine homogénéité des images affichées sur la page :

```
function reduction($nom,$fichier)
//voici la fonction qui redimensionne les images
{
$size = getimagesize("/chemin/galeries/$galerie/$fichier");
$produit= $size[0]*$size[1];
if ($produit>640000){
   $multiple=6;
}
elseif ($produit>360000){
   $multiple=5;
}
elseif ($produit>250000){
   $multiple=4;
}
elseif ($produit>160000){
   $multiple=3;
}
else {
   $multiple=2;
}
$dest_largeur = ($size[0]/$multiple);
//détermine la taille de la miniature, ici l'on désire qu'elles
soient d'un sixième de leur taille normale
$dest_hauteur = ($size[1]/$multiple);

$src_img = imagecreatefromjpeg("/chemin/$userfile_name");
$dst_img = imagecreatetruecolor($dest_largeur,$dest_hauteur);

@imagecopyresampled($dst_img, $src_img, 0, 0, 0, 0, $dest_largeur,
$dest_hauteur, $size[0], $size[1]);
// la fonction qui redimensionne les photos
imagejpeg($dst_img, "chemin/galeries/$galerie/mini_$fichier", 60);
//60 représente la qualité de l'image compressée de 1 à 100
imagedestroy($src_img);
imagedestroy($dst_img);
//on libère la mémoire
}
resize($nom, $fichier);
?>
```

Maintenant, voyons le code de la page d'affichage. Vous récupérez les données sur la table pour afficher les images avec les légendes adéquates.

▲ Figure 7.9 : *La page de la galerie photo*

Le code ressemble à ceci :

```
while($rang=mysql_fetch_array($resultat){
…
echo "<a href=\"$fichier\" target=\"_blank\" border=\"0\">
<img src=\"mini_$fichier\" height=\"$hauteur\" width=\"largeur\">
</a><br>$legende<br>":
}
```

Vous pourrez compléter l'application en ajoutant une table pour les différentes galeries contenant un code, le nom, le nom de l'auteur, sa biographie, son site… La table des images se référerait au code dans la table `galeries`.

Les données EXIF

EXIF est un module de PHP qui lit les métadonnées intégrées dans les photos numériques. EXIF est une norme pour les constructeurs d'appareils numériques qui inclut du code dans les photos. Afin de l'utiliser, il vous faut vérifier que vous disposez de ce module avec la fonction

phpinfo(). Dans notre exemple, nous allons lire la date de prise de vue, l'ouverture de l'objectif, la vitesse d'obturation et la vitesse ISO. La norme est suivie avec parfois quelques libertés, et selon l'appareil certaines données seront présentes ou non.

```
$exif=read_exif_data($fichier);
if (isset($exif['DateTimeOriginal'])) {
   $date=$exif['DateTimeOriginal'];
}
if (!$date && isset($exif['DateTime'])) {
   $date=$exif['DateTime'];
}
if ($date) {
   $date=split(':',str_replace(' ',':',$date));
   $date=mktime($date[3],$date[4],'0',$date[1], $date[2], $date[0]);
   //ici la date sera au format français
   setlocale (LC_ALL, 'fr_FR');
   $ladate=strftime ("%e %B %Y, %Hh%M", $date);

   echo $ladate."<br>";
}

if (isset($exif['COMPUTED']['ApertureFNumber'])) {
   echo $exif['COMPUTED']['ApertureFNumber']."<br>";
}
echo $exif['ExposureTime']."<br>";
if (isset($exif['ISOSpeedRatings'])) {
   $iso=$exif['ISOSpeedRatings'];
}
elseif (isset($exif['MakerNote']) && isset($exif['ModeArray'])) {
   switch (@$exif['ModeArray'][16])
   {
      case 15: $iso="auto";break;
      case 16: $iso="50";break;
      case 17: $iso="100";break;
      case 18: $iso="200";break;
      case 19: $iso="400";break;
   }
}

if ($iso) {
   echo $iso."<br>";
}
```

Ces morceaux de code constituent une base de travail sur laquelle vous pourrez construire une application personnalisée.

8

Moteurs de recherche et expressions rationnelles

Avec des bases de données, le volume de données peut être vertigineux, même si vous n'atteindrez pas celui des grands annuaires de liens comme Yahoo ou Nomade.

8.1 Un annuaire de liens

Le principe ici est de partager des liens sur un site. Chaque membre pourra ajouter des liens afin que tout le monde en profite.

Analyse du besoin

Chaque lien entré comporte plusieurs paramètres :

- l'URL ;
- le pointeur ou le texte qui apparaît comme lien ;
- la catégorie du lien ;
- la langue du lien ;
- la description du lien.

Le carnet de liens comprend deux niveaux de liens :

- les liens qui sont une composante du profil du membre intitulée *Mes liens favoris* ;
- les liens qui composent l'annuaire de liens.

L'application pose trois problèmes :

- Dans ce club de Bons Vivants, les membres possèdent des intérêts communs et risquent de proposer plusieurs fois le même site.
- Le nombre de liens ira croissant. Assez vite, il ne permettra plus de naviguer à vue. Il faudra alors un radar, c'est-à-dire un moteur de recherche.
- Au fil du temps, un certain nombre de liens vont mourir, parce que les sites auront déménagé ou n'existeront plus. Il se peut également que le nom de domaine n'ait pas été renouvelé et qu'un site pornographique ait "squatté" le nom du site.

Réflexions

Pour la plupart des paramètres du lien, la sélection est faite par le membre du club. L'internaute choisit le site, donc l'URL, la langue et la description du lien. Pour le reste, il doit déterminer la langue et la catégorie. C'est à vous de lui proposer les différentes options.

Les langues

Vous ne proposerez pas toutes les langues du monde. Nous pouvons déjà restreindre le choix aux langues européennes et prendre, parmi elles, les plus fréquentes en Europe Occidentale : français, anglais, allemand, italien, espagnol et portugais. Ces langues seront codées selon leur pays d'origine (France, Grande-Bretagne, Allemagne, Italie, Espagne et Portugal). Cela nous permettra d'utiliser un bouton avec le drapeau du pays pour le choix d'une langue.

Les catégories

La difficulté est ensuite de choisir les catégories de liens. Certains sites cumuleront plusieurs catégories. Il est vrai qu'il vaut mieux vous accorder un peu de temps pour la réflexion car de cette étape dépend la suite de votre application. Il est toujours délicat de modifier l'application pendant qu'elle fonctionne. Trouvez l'équilibre entre le désir de tout contrôler et celui de se frotter à la réalité et d'effectuer un ajustement. Vous pouvez lancer une application avec une phase de test et prévoir la possibilité pour chaque membre de modifier les caractéristiques de ses liens.

Nous allons procéder ainsi car la facilité d'ajustement d'une application est une des caractéristiques du langage PHP la plus intéressante pour un site web.

Le membre pourra à tout moment modifier les paramètres du lien.

Le problème des doublons

Si deux membres proposent le même site dans l'annuaire de liens et non dans le profil du membre, il nous faudra uniformiser les liens pour déceler les doublons. Ce tri se fera au moment de la requête SQL et de l'affichage des liens.

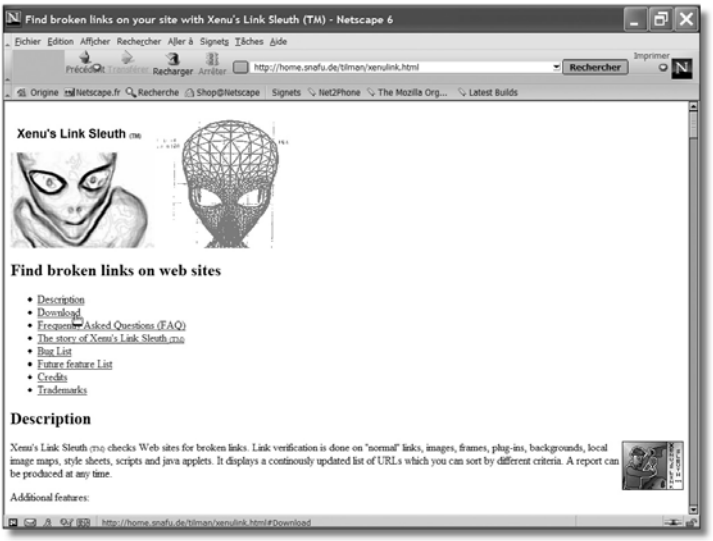
Le moteur de recherche

Le moteur de recherche sera utilisé pour les recherches en plein texte car pour les recherches par catégories, les requêtes SQL suffiront. Nous prévoyons aussi une recherche sur des mots tronqués par le moyen des étoiles (le signe astérisque). Pour créer un moteur fin et puissant, nous utiliserons les expressions rationnelles de Perl disponibles en langage PHP.

Le problème des liens morts

Pour épurer régulièrement les liens morts dans l'annuaire, il existe deux possibilités :

- Nous créons un fichier qui n'est pas destiné au public. Il contient la liste de tous les liens. Cela veut dire qu'à l'entrée d'un nouveau lien, il est recréé. Nous utilisons alors un logiciel libre comme Xenu (`http://home.snafu.de/tilman/xenulink.html`) pour scanner les liens du fichier et les analyser.



▲ Figure 8.1 : *Xenu*

- Nous créons une application qui suit chaque lien et ramène les réponses HTTP. Nous récupérons l'en-tête HTTP et son code et effaçons les liens qui génèrent une erreur.

Élaboration des applications

Pour élaborer l'annuaire, il nous faut créer le formulaire d'ajout de lien, puis l'enrichir en gardant la possibilité de modifier ou supprimer le lien. Nous arrivons ensuite à la fabrication de la page d'affichage et de la requête du moteur de recherche dans la base.

La deuxième application, sur le tableau de bord de l'administrateur du site, vérifie les liens morts et crée la liste des mots les plus souvent demandés par les visiteurs.

Création du formulaire

Pour commencer, nous créons la table des données. Nous ajouterons une table *langue* et une table *categorie*.

Voici les champs de la table *liens_annuaire* :

```
clef int(11) AUTO_INCREMENT
pointeur varchar (50)
lien varchar (70)
categorie char(1)
langue char(2)
commentaire text
moment int(8)
```

Nous créons une table *langue* composée comme suit :

```
CREATE TABLE langue (code CHAR(2) not null, nom VARCHAR(12) not null ,
PRIMARY KEY (code), INDEX (code), UNIQUE (code))
```

Ensuite, nous la remplissons ainsi :

Tab. 8.1 : Codes ISO de la table langue

Code ISO	Langue
fr	français
en	anglais

Tab. 8.1 : Codes ISO de la table langue

Code ISO	Langue
es	espagnol
de	allemand
it	italien
pt	portugais

Dans la table *categorie_liens*, bâtie sur le même modèle avec les champs *code* et *nom*, nous insérons les catégories : rencontres, producteurs, restaurants, diététique, art de vivre, paysages et voyages. Il vous suffit ensuite d'incrémenter une lettre dans le code pour chaque insertion ('A', 'B'…).

Utilisez des lettres pour le code

Nous évitons de mettre un chiffre dans le code car nous n'en avons que dix à notre disposition, en incluant le 0. Si nous ajoutons des catégories par la suite, il vaudra mieux disposer des vingt-six lettres. Si vous avez besoin de plus de code, incrémentez les lettres comme nous l'avons vu au chapitre *Les clefs du PHP* sur les opérateurs unaires. Ainsi, vous commencez une boucle à `AA` que vous incrémentez. La boucle passera en revue toutes les combinaisons à deux lettres avec un code unique.

Une fois les tables construites, intéressons-nous au formulaire. Vous allez créer le formulaire, comme auparavant, avec les colonnes des tables de la base comme champs du formulaire. N'oubliez pas les deux menus créés à partir des données des tables *langue* et *categorie*. Ensuite, créez le code Javascript et le traitement des données sur le même fichier avec une requête SQL d'insertion, en testant la présence du champ caché *$passage*. À chaque insertion, le membre recharge la page d'insertion de lien. Elle est ornée d'un message qui lui annonce que le lien précédent a été ajouté et qu'il peut en insérer un autre s'il le désire.

LAST_INSERT_ID()

`LAST_INSERT_ID()` est une fonction MySQL très pratique pour l'insertion d'une clé étrangère. Si vous utilisez la fonction `AUTO_INCREMENT` pour générer une clé unique, vous pouvez, après l'insertion de données dans une table, utiliser la dernière clé sans faire une requête `SELECT`. Une autre façon consiste à utiliser la fonction `mysql_insert_id()` après une insertion.

Nous allons détailler le traitement du formulaire mais auparavant, décortiquons la ligne contenant une expression rationnelle.

Tab. 8.2 : Le même chemin sous plusieurs formes

URL
`http://www.nh.ultranet.com/~stewoody`
`http://www.nh.ultranet.com/~stewoody/`
`http://www.nh.ultranet.com/~stewoody/index.html`
`http://www.nh.ultranet.com/~stewoody/index.htm`
`http://www.nh.ultranet.com/~stewoody/default.php`

Le navigateur va traduire l'adresse pour arriver à la page `index` ou `default`. Dans notre annuaire de liens, nous voulons éviter les doublons, c'est pourquoi il nous faut uniformiser la syntaxe des liens pour faire des comparaisons. L'expression rationnelle

```
$lien=preg_replace("#^(http://)
(.*)/((default)|(index))*(\.*)[A-z0-9]{0,4}$#", "\\1\\2", $lien);
```

recherche une ligne qui contient une chaîne de caractères commençant par `http://` et finissant par `/defaut`, `index .html`, `php3` ou `php`. On raccourcit ici la chaîne en enlevant toute barre oblique de fin et éventuellement tout nom de fichier en `index` ou `default` avec son extension. Dans le deuxième membre de la fonction, nous récupérons le contenu des deux premières parenthèses. Toute URL sous cette forme sera raccourcie pour arriver à la forme de la première ligne du tableau. Nous parlerons de ce processus plus avant dans le chapitre.

```
$moment= date ("Ymd");
$passage=$_GET['passage'] ;
if ($passage==1){
if (!ereg("http://",$lien){
$lien="http://".$lien;
///////Nous ajoutons http:// s'il manque////
}
$lien=preg_replace("#^(http://)(.*)/((default)|(index))*
(\.*)[A-z0-9]{0,4}$#", "\\1\\2", $lien);
//nous enlevons  les / ou index…
$sql="insert into liens_annuaire (pointeur, lien, categorie,
 commentaire, moment, nom_usage, langue) VALUES ('$pointeur',
 '$lien', '$categorie', '$commentaire', '$moment', '$nom_usage',
 '$langue')";
@mysql_query($sql,$id_link);
```

En haut du programme, nous avons inséré la ligne qui fournit la valeur de la variable `$moment`. Désormais, nous allons l'ajouter au fichier *fonctions.inc.php* pour l'avoir constamment à notre disposition.

Pour extraire les liens de la table, nous utiliserons la clause `DISTINCT`. Nous avons rationalisé la forme des URL avec une expression rationnelle. Les doublons auront la même forme, ce qui nous permettra de les déceler. Deux membres qui ajoutent une référence au même site mais à une page différente commettent-ils un doublon ? Non, car l'information sera certainement différente. Les pages personnelles chez un même hébergeur gratuit ont également le même nom de domaine mais des sites totalement différents. Dans un cas comme celui-ci, il vaut mieux ne pas être perfectionniste.

Date() et mktime() ou TIMESTAMP

Les deux solutions créent une date sous la forme d'un entier de type `AAAAMMJJ`. Dans les deux cas, ce nombre permet d'ordonner les dates facilement dans un ordre croissant ou décroissant. La fonction MySQL `TIMESTAMP` ajoute une date automatiquement. Son intérêt, qui devient un inconvénient dans certains cas, est qu'elle permet de modifier cette date à chaque changement de la ligne. Cette action est intéressante si vous voulez connaître la dernière date de mise à jour mais, dans le cas qui nous concerne, la date de création est plus intéressante. Nous ajoutons la date d'aujourd'hui avec les fonctions `date()` et `mktime()` car ce nombre n'est pas modifié à chaque mise à jour de champ.

Modification ou suppression du lien

Pour modifier le lien ou le supprimer, ajoutez un lien sur le tableau de bord des membres. Avant d'afficher ce lien, il faut que le membre en question ait placé des liens dans la table. Une requête SQL permet de le vérifier facilement :

```
$sql="select clef from liens_perso where nom_usage='$nom_usage'";
$resultat=@mysql_db_query($dbname,$sql,$id_link);
$nombre_rangs=mysql_num_rows($resultat);
if ($nombre_rang==1){
echo "<A HREF=\"modif_lien.php\">Modifier ou supprimer un lien</A>";
}
```

En haut du fichier *modif_lien.php*, l'authentification conserve les données du membre. La page d'affichage consiste en une requête `SELECT` et une boucle sur son résultat. Sur chaque ligne sont élaborés deux formulaires différents avec un seul bouton et un champ caché :

```
$sql="select clef, pointeur, lien from liens_perso
 where nom_usage='$nom_usage'";
$resultat=@mysql_db_query($dbname,$sql,$id_link);
while ($rang=mysql_fetch_array($resultat)){
$clef=$rang['clef'];
$pointeur=$rang['pointeur'];
$lien=$rang['lien'];
echo "<A HREF=\"$lien\" target=\"_blank\">$pointeur</A>";
echo "<A HREF=\"insérer_lien?effet=modifier&clef=$clef\"
 target=\"_blank\">modifier</A>";
echo "<A HREF=\"$PHP_SELF?effet=supprimer&clef=$clef\"
 target=\"_blank\">supprimer</A>";
}
```

Page d'affichage

Nous allons afficher les liens à la manière d'un moteur de recherche comme Google… En haut des pages, nous allons concevoir un formulaire pour choisir les critères d'affichage (langue, catégorie) et les critères de tri (alphabétique, date décroissante, nombre de liens par page). Nous insérons un champ de recherche juste avant le bouton **Envoyer**.

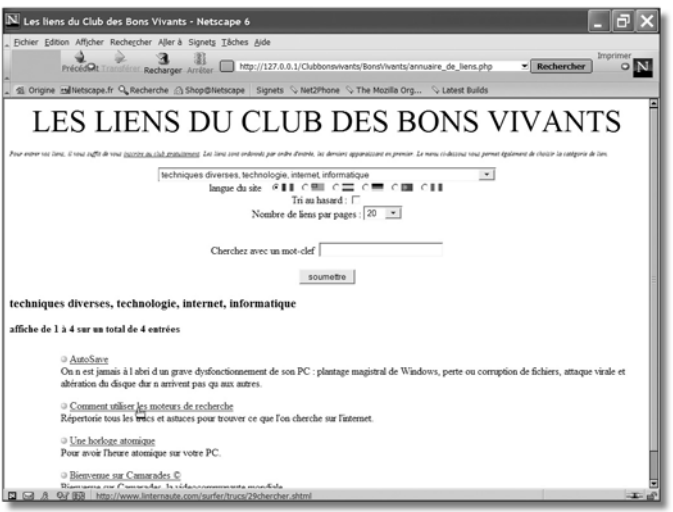
La page est constituée de la liste des liens. En haut, une ligne indique le nombre total de résultats trouvés et le nombre de liens affichés dans la page. En dessous s'affiche un lien vers la page suivante ou précédente.

Vous pouvez, bien sûr, améliorer cet affichage. Vous ajoutez, par exemple, une ligne de liens désignant chaque page suivante. La page courante serait distinguée des autres pages car elle apparaît en gras avec une absence de lien. Vous pouvez aussi créer un tableau coloré où chaque cellule contient le lien vers une page et où la page courante est désignée par une couleur différente.

L'affichage par défaut aura comme caractéristiques le français comme langue et "aucune catégorie" comme autre critère. L'ordre par défaut peut être la date décroissante, ainsi les visiteurs pourront prendre comme habitude de venir voir les dernières insertions, une fois qu'ils connaissent tous les liens. Vingt liens par page par défaut permettront un temps d'affichage rapide de la page.

Pour améliorer la connaissance des visiteurs qui viennent sur votre site, nous ajoutons une table qui indiquera le nombre de fois où un mot ou une expression est entrée dans le champ de recherche.

```
CREATE TABLE motscles (mot varchar(50) NOT NULL default '',
 fois int(10) NOT NULL default '1', PRIMARY KEY (mot),
 UNIQUE KEY mot (mot) )
```



▲ Figure 8.2 : *Annuaire de liens*

Pour l'application, si une variable n'est pas reçue, sa valeur par défaut prend le relais. Les liens vers les pages suivantes contiennent les variables en mode GET :

```
<?
include_once "../commun/fonctions.inc.php";
include_once RACINE."/commun/connexion.inc.php";
include_once RACINE."/commun/stats.inc.php";
$table='liens_annuaire';
//////////EN-TÊTE HTML/////////////////
echo '<html><head><title>Les liens du Club des Bons
Vivants</title>';
echo '<meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1">';
echo '<script language="JavaScript">';
echo 'function ouvrir_vasistas(adresse, nom_fenêtre,
caracteristiques)
 { ';
echo "\n";
echo "window.open(adresse, nom_fenêtre,caracteristiques);\n";
echo "}\n";
echo "</script>\n ";
echo "</head>\n ";
////////FIN EN-TÊTE HTML//////////////////////
echo "<body bgcolor=\"#FFFFFF\" text=\"#000000\" link=\"#0000FF\"
 vlink=\"#6600CC\" alink=\"#FF0000\">\n";
echo '<p align="center">
<font size="7">LES LIENS DU CLUB DES BONS VIVANTS</font></p>';
echo "\n";
```

Cette première partie inclut les fichiers indispensables et envoie le code HTML du haut de la page. Maintenant, construisons la requête SQL en fonction des choix de l'internaute avec un choix par défaut.

```
/*Nous vérifions si l'internaute A UTILISÉ LE FORMULAIRE pour
 changer les critères. Dans ce cas, il faut remettre l'affichage
 des liens à 0 */
if (count($_POST)>0){
$debut=0;
}
////FIN TEST SUR FORMULAIRE////////////
/* nous allons maintenant créer des VALEURS PAR DÉFAUT pour tous
 les champs qui ne sont pas renseignés par méthode GET ou POST – A
 VOUS D'AJOUTER $_GET ou $_POST*/
if (!$nbre_affiche){
$nbre_affiche=20;
}
```

```
if (!$debut){
$debut=0;
}
if (!$langue){
$langue='fr';
}
//FIN INITIALISATION DES VARIABLES SUR LEUR VALEUR PAR DÉFAUT
/*CONTRUCTION LA REQUÊTE SQL AVEC 3 VARIABLES $tri pour les
 critères d'ordre, $champ pour les critères de champ, $limites
 pour les critères de limite. Nous en profitons pour crêer les
 champs GET dans l'URL*/
//D'ABORD LA CLAUSE WHERE/////////////////////////
/*ici il est possible d'accéder aux liens personnels entrés par
 un des membres en particulier*/
if (strlen($nom_annuaire)){
  $champ="where nom_usage LIKE '%$nom_annuaire%'";
  $traine="&nom_annuaire=$recherche";
  }
```

Les deux variables construites sont `$champ` qui aide à l'élaboration de la requête SQL au niveau des clauses et `$traine` qui complète le lien qui servira à transporter la requête vers les pages suivantes ou précédentes.

```
//////////////////LE MOTEUR DE RECHERCHE///////////
elseif(strlen($mot_clef)){
 $champ="where pointeur like '%$mot_clef%' OR
 contenu like '%$mot_clef%'";
 $traine="mot_clef=$mot_clef";
}
//////SINON, LE CHAMP LANGUE SERA TOUJOURS PRÉSENT//
else {
$champ ="where langue='$langue'";
$traine="&langue=$langue";
if (strlen($categorie) && $categorie!=0){
  $champ.=" AND categorie='$categorie'";
  $traine.="&categorie=$categorie";
  }
 }
/////////////ORDRE TEMPOREL DÉCROISSANT OU HASARD////////
if ($ordre=='hasard'){
$nouvel_ordre='ORDER BY RAND()';
$traine.="&ordre=$hasard";
}
else {
$nouvel_ordre ='ORDER BY clef DESC';
}
?>
<font size="-2"><i>Pour entrer vos liens, il vous suffit de vous
 <a href="http://bonsvivants.com/club/">inscrire au club
```

```
 gratuitement</a>. Les liens sont ordonn&eacute;s par ordre
 d'entr&eacute;e, les derniers apparaissant en premier.
 Le menu ci-dessous vous permet &eacute;galement de choisir
 la cat&eacute;gorie de lien.</i></font><p> <div align="center">
<?
//LE FORMULAIRE///////////////
```

Nous construisons le formulaire qui précise les choix de l'internaute à l'aide des champs pour la catégorie de lien, la langue, le nombre de liens par page et les mots-clé recherchés. Une option supplémentaire permet de procéder à un affichage aléatoire.

```
   $sql="select clef FROM categories_liens";
$resultat=mysql_query($sql,$id_link);
$nombre=mysql_num_rows($resultat);
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
      echo "<select name=\"categorie\" size=\"1\">";
  echo "<option value=\"0\">choisissez parmi les $nombre</option>";
   $sql="select clef, nom FROM categories_liens";
$resultat=mysql_query($sql,$id_link);
while ($rang=mysql_fetch_array($resultat)){
$nom=$rang['nom'];
$clef=$rang['clef'];
$nom=$rang['nom'];
if ($categorie>0 && $categorie==$clef){
 $select=' SELECTED';
   }
   else {
   $select ='';
   }
 echo "<option value=\"$clef\"";
 echo $select;
 echo ">$nom</option>";
}
echo "</select>";
?>
<br>langue du site
<input type="radio" name="langue" value="fr"
<?
if ($langue=='fr' || !isset($langue)){
echo " checked";
}
echo ">";
?>
<img src="drapeaux/france.gif" width=18 height=12 border=0>
  <input type="radio" name="langue" value="en"
<?
if ($langue=='en'){
```

```
echo " checked";
}
echo ">";
?>
<img src="drapeaux/usa.gif" width=20 height=12 border=0>
  <input type="radio" name="langue" value="es"
<?
if ($langue=='es'){
echo " checked";
}
echo ">";
?>
<img src="drapeaux/spain.gif" width=18 height=12 border=0>
  <input type="radio" name="langue" value="de"
<?
if ($langue=='de'){
echo " checked";
}
echo ">";
?>
<img src="drapeaux/germany.gif" width=18 height=12 border=0>
  <input type="radio" name="langue" value="po"
<?
if ($langue=='po'){
echo " checked";
}
echo ">";
?>
<img src="drapeaux/portugal.gif" width=18 height=12 border=0>
  <input type="radio" name="langue" value="it"
<?
if ($langue=='it'){
echo " checked";
}
echo ">";
?>
<img src="drapeaux/italy.gif" width=18 height=12 border=0>
<?
echo "<BR>\n";
echo "Tri au hasard : \n";
echo '<input type="checkbox" name="ordre" value="hasard"';
if ($ordre=='hasard'){
echo " CHECKED";
}
echo '><BR>';
echo "\nNombre de liens par pages : <select
name=\"nbre_affiche\">\n";
$nombres=array ('20', '30', '50', '75', '100');
foreach ($nombres as $valeur){
echo "<option value=\"$valeur\"";
```

```
if ($nbre_affiche==$valeur){
echo " SELECTED";
}
echo ">";
echo "$valeur</option>";
}
//la valeur choisie est affichée dans le menu avec selected
echo "\n</select>";
echo '</div><br><div align="center">Cherchez avec un mot-clef
 <input type="text" size="30" name="mot_clef"></div>';
echo '<br><center><input type="submit" value="soumettre">
</center></form>';
///////////////FIN DU FORMULAIRE/////
```

La catégorie est affichée comme un titre :

```
////ICI NOUS AFFICHONS EN GRAND LA CATÉGORIE//////
if (strlen($categorie)){
$sql="select nom FROM categories_liens where clef=$categorie";
$resultat=@mysql_query($sql,$id_link);
$rang=@mysql_fetch_array($resultat);
$nom=$rang['nom'];
echo "<h3>$nom</h3>";
}
```

La requête nommée `$sql_total` retourne le total de tous les liens demandés une fois les doublons éliminés :

```
$sql_total = "select count(DISTINCT lien) as nombre_total
 from $table $champ";
//nous enlevons les doublons du compte avec DISTINCT
  $resultat = mysql_query($sql_total, $id_link);
  $rang = @mysql_fetch_array($resultat);
  $nombre_total = $rang["nombre_total"];
```

Voici la vraie requête pour la page. Elle ramène les données en fonction des paramètres demandés par l'internaute :

```
///ICI NOUS ALLONS CHERCHER LES DONNÉES///
    $sql = "select DISTINCT lien, clef, pointeur, categorie,
 contenu, nom_usage, moment, langue from $table $champ
 $nouvel_ordre limit $debut, $nbre_affiche";
  $resultat = mysql_query($sql, $id_link);
  $nombre_total_page_courante =  @mysql_num_rows($resultat);
  if ($nombre_total < 1) {
    echo "<p>Il n'y a pas d'enregistrements dans cette page
 de liens.</p>";
```

```
  }
  else {
//déterminons maintenant le début et la fin de la prochaine page///
    $prochain_index = $debut + $nbre_affiche;
 if ($prochain_index > $nombre_total) {
   $prochain_index = $nombre_total;
 }
/*nous affichons la position de la page dans le total des
 enregistrements */
 $premier = $debut + 1;
    echo "<p><b>affiche de $premier à $prochain_index </b>";
 echo "<b>sur un total de $nombre_total entrées</b></p>";
//Déterminons le début de la page précédente
    $precedent_index = $debut - $nbre_affiche;
///à condition de ne pas être avant le début (0) ///
    if ($precedent_index < 0) {
     $precedent_index = 0;
    }
  }
?>
  <ul>
  <table border="0">
  <tr>
   <table width="100%"><tr>
     <td align="left" width="50%">
  <?
    //AFFICHE UN LIEN VERS LA PAGE PRÉCÉDENTE
    if ($debut != 0) {
        $commence = "debut=$precedent_index";
   echo "<a href=\"$_SERVER['PHP_SELF']?$commence$traine\">";
   echo "<b>Les $nbre_affiche précédents</b></a>";
   }
 ?>
     </td>
     <td align="right" width="50%">
  <?php
    /// AFFICHE LE LIEN VERS LA PROCHAINE PAGE///////
    if ($prochain_index != $nombre_total) {
      $finit = "debut=$prochain_index";
           echo "<a href=\"$_SERVER['PHP_SELF']?$finit$traine\">";
$prochain=$nombre_total-($debut+$nbre_affiche);
if ($prochain>$nbre_affiche){
echo "<b>Les $nbre_affiche prochains</b></a>";
}
else {
if ($prochain!=1 && $prochain>0){
echo "<b>Les $prochain prochains</b></a>";
}
elseif ($prochain==1) {
```

```
echo "<b>La dernière entrée</b></a>";
}
    }
    }
 ?>
```

Ce qui précède concerne la construction des liens vers les autres pages et leur affichage. Nous arrivons maintenant au corps de la page qui affiche la liste des liens. En bas de la page, nous affichons à nouveau les liens vers les autres pages pour le confort de l'internaute :

```
     </td>
   </tr></table>
  </td></tr>
  <tr><td >
<UL>
    <?
//////////////////////
    for ($index = 0;
      $index < $nombre_total_page_courante;
   $index++) {
      $rang = @mysql_fetch_array($resultat);
     $clef=$rang['clef'];
  $pointeur=$rang['pointeur'];
  $pointeur=stripslashes($pointeur);
    $nom_usage=$rang['nom_usage'];
    $lien=$rang['lien'];
    $categorie=$rang['categorie'];
    $contenu=$rang['contenu'];
    $contenu=stripslashes($contenu);
    $nom_usage=$rang['nom_usage'];
echo "<img src=\"puces/${categorie}.gif\" width=10 height=10
border=0>
 <a href=\"$lien\">$pointeur</a><br>$contenu<p>";
    }
 ?>
 </ul>

  </td></tr>  <tr><td>
   <table width="100%"><tr>
     <td align="left" width="50%">
  <?
    //AFFICHE UN LIEN VERS LA PAGE PRÉCÉDENTE
    if ($debut != 0) {
        $commence = "debut=$precedent_index";
   echo "<a href=\"$_SERVER['PHP_SELF']?$commence$traine\">";
   echo "<b>Les $nbre_affiche précédents</b></a>";
   }
```

```
 ?>
     </td>
     <td align="right" width="50%">
  <?php
    /// AFFICHE LE LIEN VERS LA PROCHAINE PAGE///////
    if ($prochain_index != $nombre_total) {
      $finit = "debut=$prochain_index";
            echo "<a href=\"$_SERVER['PHP_SELF']?$finit$traine\">";
$prochain=$nombre_total-($debut+$nbre_affiche);
if ($prochain>$nbre_affiche){
echo "<b>Les $nbre_affiche prochains</b></a>";
}
else {
if ($prochain!=1 && $prochain>0){
echo "<b>Les $prochain prochains</b></a>";
}
elseif ($prochain==1) {
echo "<b>La dernière entrée</b></a>";
}
    }
    }
 ?>
     </td>
   </tr></table>
  </ul>
<body></html>
```

La page se termine par une requête SQL qui entre les mots-clé demandés. Cette table vous donnera une bonne indication sur les centres d'intérêt de vos visiteurs.

Listing 8.1 : *Affichage des liens sur plusieurs pages*
annuaire_de_liens.php

```
<?
/* POUR FINIR ON ENTRE LE MOT CLEF ÉVENTUEL DANS LA TABLE mots_clef*/
if (strlen($mot_clef)){
$sql="select * from mots_clef where mot='$mot_clef'";
$resultat=mysql_query($sql, $id_link);
$nombre=mysql_num_rows($resultat);
if ($nombre>0){
while($rang=mysql_fetch_array($resultat)){
$fois=$rang['fois'];
$fois++;
$str_update="UPDATE mots_clef SET fois='$fois' where
mot='$mot_clef'";
  mysql_query($str_update, $id_link);
}
}
```

```
else {
$sql="INSERT INTO mots_clef (mot) VALUES ('$mot_clef')";
  mysql_query($sql, $id_link);
}
}
?>
```

Ce type d'application vous servira constamment. Le principe est d'afficher les données dans un ordre quelconque sur plusieurs pages. Les bases de données gèrent des masses de données et ce type d'application est indispensable pour la plupart des pages qui affichent des listes.

Vérification des liens

Pour cette vérification, nous serons pragmatique. Théoriquement, il est possible de vérifier les liens avec des fonctions PHP mais cela prend parfois beaucoup de temps et donne un résultat incertain. Pour une page de liens, il vaut mieux employer un logiciel gratuit qui permet une interrogation de plusieurs liens à la fois. Pour l'utiliser, nous créons un fichier qui contient tous les liens et un formulaire pour enlever les liens invalides.

Dans la page de traitement de l'insertion des liens que nous avons créée précédemment dans ce chapitre, nous ajoutons un peu de code :

```
$contenu =' <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
 <title>Page de vérification de liens</title>
</head>
<body>
<H2>Vérification des liens</H2><UL>';
$sql="select clef, pointeur, lien from liens_annuaire";
$resultat=mysql_query($sql, $id_link);
while ($rang=mysql_fetch_array($resultat)){
contenu.= "<LI><A HREF=\"$lien\">$pointeur</A>   $clef\n";
}
$contenu."/UL>\n</body>\n</html>";
$fd_txt=@fopen('secret/tousleliens.htm','w');
@fputs ($fd_txt,$contenu, strlen ($contenu_));
@fclose($fd_txt);
```

Le fichier sera écrasé à chaque nouveau lien et reconstruit. Vous le mettez dans vos coulisses, par exemple un répertoire *secret* à qui vous donnez les permissions adéquates et qui reste à l'écart des robots parcourant le Web.

Un formulaire peut l'accompagner.

```
Veuillez entrer les clefs des liens à effacer :<BR>
<?
echo "<form action=\"$_SERVER['PHP_SELF'] \" method=\"post\">\n";
?>
Les clefs des entrées à effacer, séparées par des &
<input type="text" name="clefs" size="50">
<input type="submit" value="envoyez">
</form>
```

En haut du fichier, vous ajoutez :

```
<?
include "../commun/fonctions.inc.php";
include RACINE."/commun/connexion.inc.php";
$lesclefs=explode('&', $clefs);
foreach ($lesclefs as $val){
$val=trim($val);
$sql="delete from liens_annuaire where clef='$val'";
mysql_query($sql, $id_link);
}
?>
```

Robots.txt, un chien de garde pour les moteurs de recherche

Dans le répertoire *secret*, vous placez les fichiers secrets qui doivent rester accessibles sur le Web pour vous. Aucun lien n'y mène. Ils sont gardés par un fichier nommé *robots.txt* qui se trouve dans le répertoire *www* ou *htdocs*. Ce fichier est un fichier texte dans lequel vous inscrivez :

```
User-agent: *
Disallow: /secret/
Disallow: /tmp/
Disallow: /coulisses/index.php
```

L'exemple précédent demande aux robots fouineurs de ne pas indexer les répertoires *secret*, *tmp* et le fichier *index.php* dans le répertoire *coulisses*. Chaque fois qu'un robot visite un site, il commence par chercher ce fichier pour savoir ce qui ne doit pas être indexé dans le répertoire web.

Moteur de recherche

Nous allons maintenant étudier le principe d'un moteur de recherche en prenant comme cible les pages du magazine. Le principe est le même que celui de l'annuaire de liens, une page d'affichage avec un formulaire pour entrer les mots recherchés.

Recherche dans la table MySQL

Avant de traiter le formulaire, il faut construire la requête à partir de mots entrés et placer les résultats dans une table transitoire. Nous emplirons cette table avec les résultats puis la viderons.

Cette table est la même table dupliquée plusieurs fois dont les noms sont *recherche0*, *recherche1*… Avant d'utiliser l'une de ces tables, le programme cherchera un nombre sur un compteur qu'il incrémente ou remplace par 0 s'il est égal à 10.

Tables temporaires

À partir de la version 3.23.02, MySQL dispose de la possibilité de créer des tables temporaires. Cette astuce des tables transitoires vaut pour les versions antérieures de MySQL.

En haut du fichier, nous avons cette requête :

```
if (!$table){
$sql="select compteur from tables_temporaires";
$resultat= @mysql_query($sql,$id_link);
$rang=mysql_fetch_array($resultat);
$compteur=$rang['compteur'];
$table='recherche'.$compteur;
$compteur++;
if ($compteur==10){
$compteur='0';
}
$sql="update tables_temporaires set compteur='$compteur'";
@mysql_query($sql,$id_link);
}
```

Créons un formulaire pour entrer les requêtes des abonnés du magazine. Le champ comprend jusqu'à trois mots séparés par un espace.

```
<?
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
?>
Pour votre recherche, vous pouvez proposer jusque 3 mots
 séparés par un espace
<input type="text" name="touslesmots" size="50">
<input type="submit" value="Recherchez">
</form>
```

Créons un formulaire pour entrer les requêtes des abonnés du magazine. Le champ comprend jusqu'à trois mots séparés par un espace.

```
<?
echo "<form action=\"$_SERVER['PHP_SELF']\" method=\"post\">";
?>
Pour votre recherche, vous pouvez proposer jusque 3 mots
 séparés par un espace
<input type="text" name="touslesmots" size="50">
<input type="submit" value="Recherchez">
</form>
```

Créons maintenant un tableau pour entreposer les mots :

```
$dico=explode(" ",$touslesmots);
$combien=count($dico);
if ($combien==1){
$sql[]="select chemin_complet from magazine where article
 like '%$tab[0]%'";
}
elseif ($combien==2){
$sql[]="select chemin_complet from magazine where article
 like '%$tab[0]%' AND article like '%$tab[1]%'";
$sql[]="select chemin_complet from magazine where article
 like '%$tab[0]%' OR article like '%$tab[1]%'";
}
elseif ($combien==3){
$sql[]="select chemin_complet from magazine where article like
 '%$tab[0]%' AND article like '%$tab[1]%' AND article like
 '%$tab[2]%'";
$sql[]="select chemin_complet from magazine where article
 like '%$tab[0]%' AND article like '%$tab[1]%'";
$sql[]="select chemin_complet from magazine where article
 like '%$tab[0]%' AND article like '%$tab[2]%'";
$sql[]="select chemin_complet from magazine where article
 like '%$tab[1]%' AND article like '%$tab[2]%'";
$sql[]="select chemin_complet from magazine where article
 like '%$tab[0]%' OR article like '%$tab[1]%' OR article like
 '%$tab[2]%'";
```

```
}
foreach($sql as $val){
$sql_select=$val;
$resultat=@mysql_query($sql_select,$id_link);
while($rang=mysql_fetch_array($resultat)){
$chemin_complet=$rang['chemin_complet'];
$sql_insert="insert into $table (chemin_complet)
 VALUES ('$chemin_complet')";
mysql_query($sql_insert,$id_link);
}
}
```

Une fois cette table remplie avec les résultats, il vous suffit d'afficher les données. Vous avez sans doute compris le principe de ce moteur de recherche. Nous avons simplifié mais vous pouvez le rendre plus complexe avec une recherche sur plusieurs colonnes de la table. Si vous augmentez le nombre de mots, la complexité augmente proportionnellement.

Indexation de tous les mots dans chaque fichier

Vous pouvez également indexer les mots de chaque texte et, un peu comme les compteurs de visites, compter le nombre de fois où les mots apparaissent dans un texte. Vous prenez soin d'en exclure tous les mots ou signes non significatifs comme les articles, les prépositions, les pronoms, la ponctuation et les balises avec des expressions rationnelles. Nous allons étudier le raisonnement, les expressions rationnelles et les fonctions principales qui nous permettront de réussir ce moteur.

Les fichiers indexés seront uniquement les nouveaux fichiers quand ils seront validés pour la parution. Le rédacteur en chef déclenchera lui-même l'indexation avec son tableau de bord. Le moteur d'indexation va en recueillir les contenus, les épurer à l'aide d'expressions rationnelles, en calculer la fréquence puis insérer ces données dans une table *MySQL*.

C'est la fréquence d'un mot dans un texte qui décide de sa place dans les résultats. Si un mot se trouve dans le titre, dans la description de la page ou dans ses mots-clé, il prend une valeur plus importante.

Pour le répertoire du numéro 4 du magazine, nous indexons les fichiers (les images sont toutes dans un répertoire différent).

```
$rep='quatre';
$dir = opendir("/$rep");
while ($f = @readdir($dir)) {
```

```
if(!is_dir($f)){
$chemin_fichier="quatre/$f";
```

Nous récupérons les balises `meta` qui sont en tête des fichiers HTML et contiennent les mots-clé ainsi que la description de la page avec la fonction `get_meta_tags()` :

```
$titres=get_meta_tags( $f);
$titre=$tableau_meta['keywords'];
$titre.=" ".$tableau_meta['description'];
```

Nous les joindrons au titre pour leur donner de l'importance.

```
$fd_txt=@fopen($f,'r');
$contenu_txt = @fread ($fd_txt, filesize ($f));
@fclose($fd_txt);
```

Pour extraire le titre, nous utilisons une expression rationnelle.

```
preg_match ("/(<(title)[^>]*>)+?((.|\n)*?)
(<\/\\2>)/i", $contenu_txt, $trouvailles);
$titre.= " ".$trouvailles[3];
```

C'est une expression de type Perl avec `preg_match()` que nous étudierons à la fin du chapitre. Elle sert à capturer les mots à l'intérieur des balises <title> ou <TITLE> et </title> ou </TITLE>.

Nettoyons maintenant le texte de ce qui est inutile : ponctuation, mots composés de moins de trois lettres et balises. Nous pourrions préciser également un certain nombre de mots inutiles.

Avec ces deux fonctions, nous transformons les caractères spéciaux HTML (comme `é` en é) :

```
$codage = get_html_translation_table(HTML_ENTITIES);
$codage= array_flip($codage);
// les clefs deviennent des valeurs et vice versa //
$contenu_txt = strtr($contenu_txt, $codage);
```

Enlevons la ponctuation :

```
$contenu_txt=ereg_replace("[[:punct:]]", " ", $contenu_txt);
```

Il faudra faire des exceptions pour les mots comme "aujourd'hui" car ce mot serait divisé en deux.

Enlevons les balises

```
$contenu_txt=strip_tags($contenu_txt);
```

et les mots de moins de trois lettres :

```
$contenu_txt =ereg_replace("([[:space:]]|^)[A-zO-9]{1,2}[[:space:]]",
 " ",$contenu_txt);
```

ainsi que les doubles espaces :

```
$contenu_txt=ereg_replace("[[:space:]]{2}", " ",$contenu_txt);
```

Auparavant, faites un tableau de tous les mots inutiles (les, des, donc, qui…) et une boucle sur chaque élément avec la fonction `ereg_replace()`.

Traitez ensuite `$titre` comme vous venez de traiter `$contenu_txt` pour compter la fréquence des mots :

```
$frequence_plus=15;
$tableau_de_titres= explode(" ", $contenu_txt);
foreach($tableau_de_titres as $val){
for ($i=0;$i<$frequence_plus;$i++){
/* choisissez la valeur en fonction de la longueur des fichiers,
 tout est relatif ici,nous avons choisi 15 */
array_push ($tableau_de_titres,$val)
}
//nous ajoutons 15 fois chaque mot dans le tableau
$tableau_de_mots=explode(" ", $contenu_txt);
$tableau_de_mots=array_merge ($tableau_de_mots, $tableau_de_titres)
$tableau_des_frequences=array_count_values ($tableau_de_mots);
```

Faites ensuite un tableau de tous les mots :

```
foreach ($tableau_des_frequences as $cle=>$val){
$sql="insert into atomes (chemin_fichier, mot, fois)
 VALUES ('$chemin_fichier', '$cle', '$val')";
@mysql_query($sql,$id_link);
}
```

Enfin, pour afficher les résultats, servez-vous de ce que vous avez appris avec l'annuaire de liens. La requête SQL est la suivante :

```
$sql="select chemin_fichier, mot, MAX(fois) as maximum from atome
 where mot='$mot' GROUP BY chemin_fichier ORDER BY maximum DESC";
```

Nouvelles fonctions

array_count_values()

Retourne un tableau avec comme clés les valeurs d'un tableau et leur fréquence comme valeurs

Syntaxe : `array_count_values ( array tableau)`

get_meta_tags()

Retourne un tableau formé des balises `meta` du fichier, les clés seront les noms des paramètres et les valeurs, leur contenu

Syntaxe : `get_meta_tags ( string fichier)`

array_flip()

Intervertit clés et valeurs au sein d'un tableau

Syntaxe : `array array_flip(tableau)`

get_html_translation_table(TABLE);

Retourne le tableau des transcriptions HTML des lettres spéciales (cédille, accents…)

Syntaxe : `get_html_translation_table(TABLE [, int type_guillemets])`

Options La table peut être `HTML_ENTITIES` ou `HTML_SPECIALCHARS`

Les types de guillemets :

- `ENT_COMPAT` (par défaut), convertit les guillemets doubles mais pas les guillemets simples.
- `ENT_QUOTES`, convertit les deux : " devient `&quot`; tandis que ' devient `&#039`.
- `ENT_NOQUOTES` les laisse intacts.

8.2 Les expressions rationnelles

Dans le chapitre précédent, nous avons utilisé des expressions rationnelles pour créer le nom de la table dans le programme *mini-forum.php*.

Dans l'expression suivante, nous avons extrait le nom de fichier du chemin que nous avons écourté de son extension :

```
preg_match("#^/.*/(.*)\.[A-z0-9]{3,4}$#", $_SERVER['PHP_SELF'], $tableau);
```

Ces signes vous apparaissent sans doute comme des hiéroglyphes mais nous espérons qu'à la fin de ce chapitre, vous percevrez le langage particulier des expressions rationnelles ou régulières. Nous utiliserons l'adjectif « rationnelle » pour caractériser ces expressions car elles sont plutôt irrégulières dans le sens où elles ne régulent rien et sont utilisées différemment pour chaque cas. En fait, « régulière » est la traduction de *regular* qui dans l'expression « regular expression » signifie « expression consacrée ». Nous emploierons donc le terme « expressions rationnelles » car ces expressions permettent une recherche générique ou abstraite sur un certain nombre de caractères ou de groupes de caractères. Le type de caractère et sa place sont traduits sous forme de caractères de description. Nous avons donc traduit, sous une forme abstraite, un mot ou une expression recherchée. Ce langage est puissant mais également très complexe. Nous n'en ferons pas le tour dans ce livre, nous vous aiderons seulement à comprendre certains principes et vous en proposerons les expressions les plus courantes.

La méthode

Vu le comportement très complexe des moteurs d'expressions rationnelles, l'approche ne peut être qu'empirique. Avec un peu d'habitude, la solution sera trouvée plus rapidement. Dans chaque recherche, vous avez un objectif. Vous voulez vérifier, extraire ou remplacer une chaîne de caractères, c'est le centre de votre stratégie. Il faut aider le moteur à trouver des repères autour de cet objectif. Si vous recherchez `www`, vous savez qu'il est entouré de `http://` et d'un point. Plus vous caractérisez votre objectif, plus vous aurez de chances de le trouver. Le moteur a quelques « défauts ». Ainsi ne s'arrêtera-t-il pas au premier motif trouvé mais ira jusqu'à la fin de la cible. C'est ce qu'on appelle l'avidité. Elle

peut pousser le moteur à être trop perfectionniste dans sa recherche et à ne pas trouver ce qu'il a "sous les yeux" car il cherche toujours un motif complémentaire.

Tab. 8.3 : Début et fin de ligne

Signe	Description
^	Début de ligne, sauf quand le signe se trouve à l'intérieur de crochets. Il signifie alors une négation.
$	Fin de ligne

Les parenthèses permettent de délimiter des sous-motifs qui seront stockés dans un tableau, dans le cas d'une recherche ou dans les variables spéciales \\0 pour la totalité du motif, \\1 pour la première parenthèse…

En ce qui concerne les expressions rationnelles, PHP a hérité du module du langage Perl et des normes POSIX. Nous allons étudier un cas assez simple pour commencer puis verrons la syntaxe et approfondirons. Auparavant, nous ferons le panorama des fonctions du langage PHP qui utilisent les expressions rationnelles.

POSIX

POSIX est une norme qui permet la portabilité sur tous les systèmes.

Voici les fonctions du langage PHP aux normes POSIX.

Tab. 8.4 : Fonctions pour des expressions rationnelles POSIX

Fonction	Description
`int ereg ( string motif, string chaîne [, array tableau])`	Retourne Vrai si la chaîne contient le motif.
`string ereg_replace ( string motif, string nouveau_motif, string chaîne)`	Remplace le motif à l'intérieur de la chaîne par le nouveau motif.
`eregi()`	Comme `ereg()` mais insensible à la casse

Tab. 8.4 : Fonctions pour des expressions rationnelles POSIX

Fonction	Description
`eregi_replace()`	Comme `ereg_replace()` mais insensible à la casse
`array split ( string motif_séparateur, string chaîne [, int limite])`	Découpe la chaîne en sous-chaînes au moyen du séparateur.
`string sql_regcase ( string chaîne)`	Crée une expression rationnelle insensible à la casse à partir de la chaîne. Pour PHP, le résultat sera [Pp][Hh][Pp].

Quelques études de cas

Vous voulez savoir si vous avez des chiffres dans un texte.

```
$expression="fs5dqfdsf";
if (ereg('[0-9]',$expression)){
}
```

L'expression est vraie car elle contient au moins un chiffre. Ici `[0-9]` représente la classe `chiffres` qui pourrait aussi être exprimée avec l'expression `[[:digit:]]`. Le signe "`-`" représente l'intervalle. Il s'agit de tous les chiffres inclus dans l'intervalle 0 à 9.

Si vous voulez rechercher un chiffre particulier :

- `ereg('0',$expression)` recherche le chiffre 0.
- `ereg('[09]',$expression)` recherche les chiffres 0 ou 9 dans l'expression.

Si vous souhaitez que ces chiffres ne soient pas dans l'expression, employez le caractère ^ dans les crochets. Si vous l'employez en dehors des crochets, il n'a plus même sens.

`ereg('[^0]',$expression)` sera vrai si le chiffre 0 n'est pas dans l'expression.

Rappelez-vous :

```
if !ereg('[^0-9]',$table){
$table.='_';
```

Nous vérifiions que le nom de la table créée à partir d'un nom de fichier ne contient pas uniquement des chiffres. Si c'était le cas, nous aurions ajouté un blanc souligné.

```
$contenu_txt =ereg_replace("([[:space:]]|^)[A-z0-9]{1,2}[[:space:]]",
  " ",$contenu_txt);
```

Dans cette expression, nous cherchons un mot (entouré d'espaces) qui contienne de une à deux lettres. `([[:space:]]|^) [A-z0-9]{2,}` signifie qu'on recherche au moins deux caractères alphanumériques placés en début de chaîne ou après un espace.

Il reste un caractère que nous n'avons pas vu, c'est le caractère point (`.`). Il remplace n'importe quel nombre et type de caractère. Si vous voulez représenter un vrai point, il vous faudra utiliser un caractère d'échappement (`\.`).

Voici les expressions POSIX et leurs équivalents.

Tab. 8.5 : Les normes POSIX

Séquence	Équivalent	Description
`[[:alnum:]]`	`[A-Za-z0-9]`	Caractères alphanumériques
`[[:alpha:]]`	`[A-Za-z]`	Caractères alphabétiques
`[[:digit:]]`	`[0-9]`	Caractères numériques
`[[:blank:]]`	`[\x09]`	Espaces ou tabulations
`[[:lower:]]`	`[a-z]`	Caractères en bas de casse
`[[:upper:]]`	`[A-Z]`	Caractères en capitales
`[[:xdigit:]]`	`[0-9a-fA-F]`	Caractères hexadécimaux
`[[:punct:]]`	`[!-/:-@[-'{-~]`	Caractères de ponctuation
`[[:space:]]`	`[\t\v\f]`	Tout caractère d'espace
`[[:cntrl:]]`	`[\x00-\x19\x7F]`	Caractères de contrôle
`[[:graph:]]`	`[!-~]`	Caractères affichables et imprimables
`[[:print:]]`	`[-~]`	Caractères imprimables sauf caractères de contrôle

Les caractères sont dans l'ordre du tableau de caractères ASCII. Ainsi, dans `[a-z]`, les lettres accentuées ne figurent pas.

Les lettres accentuées

Pour ce cas , utilisez les caractères `\x` suivis du numéro dans la table *ASCII*. Pour trouver si votre texte contient des lettres accentuées françaises, vous devez écrire `[\x128-\x151\153\154]`, c'est-à-dire les caractères 128 à 151 puis les 153 et 154. De la même façon, si vous voulez le caractère littéral ^, utilisez `\x94`.

Pour affiner notre recherche, nous avons à notre disposition la cardinalité, c'est-à-dire que nous pouvons sélectionner le nombre de caractères :

- `ereg('[0-9]{1}',$expression)` recherche s'il existe un chiffre dans l'expression.
- `ereg('[0-9]{1,3}',$expression)` recherche s'il existe entre un et trois chiffres dans l'expression.

Nous avons aussi des équivalents.

Tab. 8.6 : Cardinalité

Signe	Équivalent	Description
*	{0,}	0 ou plus
+	{1,}	1 ou plus
?	{0,1}	0 ou 1

- `ereg('[0-9]+',$expression)` recherche s'il existe au moins un chiffre dans l'expression.

En voyant le tableau précédent, vous pouvez vous interroger sur l'utilité de l'étoile *. Ce signe renvoie en effet Vrai que le caractère existe ou n'existe pas. En fait, il s'emploie dans des expressions plus complexes.

- `ereg('[0-9]*',$expression)` recherche s'il existe au moins zéro chiffre dans l'expression.

Vous pouvez par exemple rechercher un mot qui peut être au pluriel ou au singulier. `!ereg('régulières*', $expression)` sera vrai si le mot "régulière" ou "régulières" n'existe pas dans l'expression ;=).

Si vous recherchez le mot événement qui peut aussi s'écrire évènement, vous emploierez l'expression `ereg('évé|ènement*', $expression)`. Le signe | signifie ou. Cette expression est équivalente à : `ereg('év[éè]nement*', $expression)`.

Vous pouvez aussi faire porter la précision sur plusieurs caractères. Ainsi, vous cherchez un mot masculin ou féminin comme docteur ou docteresse `'docteu*r(esse)*'`. Le problème est que docteuresse fonctionne aussi. `'docteu*r'` sera mieux adapté. Nous verrons un peu plus loin les problèmes d'avidité qui font que dès que l'une des expressions est trouvée, le moteur retourne Vrai sans aller jusqu'à la fin de l'expression.

Reprenons les fonctions `date_nombre()` et `nombre_date()` avec les expressions rationnelles. Le code sera plus court et l'exécution plus rapide :

```
<?
function nombre_date($cettedate){
$ladate=$cettedate;
$longueur=strlen($ladate);
if ($longueur==8){
$ladate=ereg_replace("(^[0-9]{4})([0-9]{2})([0-9]{2})","\\3/\\2/\\1",
$ladate);
}
elseif ($longueur==12){
$ladate=ereg_replace("(^[0-9]{4})([0-9]{2})([0-9]{2})([0-9]{2})",
"\\3/\\2/\\1 \\4h\\5", $ladate);
}
elseif ($longueur==14){
$ladate=ereg_replace("(^[0-9]{4})([0-9]{2})([0-9]{2})
([0-9]{2})([0-9]{2})","\\3/\\2/\\1 \\4h\\5mn\\6", $ladate);
}
if ($longueur>8){
$heure= (int)substr($ladate,8,2);
$minutes= substr($ladate,10,2);
$ladate.=" ${heure}h ${minutes}mn";
if ($longueur>12){
$secondes= substr($ladate,12,2);
$ladate.=" $secondes";
}
}
```

```
return $ladate;
}
```

Nous vérifions dans cette fonction la longueur de la date et, selon la taille, en extrayons aussi l'heure, les minutes et les secondes. Les jours, mois et heure sont forcés en `int` (typées) car nous voulons éviter qu'ils commencent par un 0.

```
<?
function date_nombre($cettedate){
$ladate=$cettedate;
$ladate=explode('/',$ladate);
$jour=$ladate[0];
$mois=$ladate[1];
$an=$ladate[2];
$longueur=strlen($an);
if ($longueur==6){
if ($an>date("y")){
$an='19'.$an;
}
else {
$an='20'.$an;
}
}
$longueur=strlen($mois);
if ($longueur==1){
$mois='0'.$mois;
}
$ladate=$an.$mois.$jour;
return $ladate;
}
?>
```

Nous voyons cette même recherche de motif à la manière de Perl dans les pages qui suivent.

Les fonctions compatibles Perl

L'intérêt majeur de ces fonctions compatibles avec le langage Perl est que vous pouvez utiliser des options utiles, voire indispensables.

Tab. 8.7 : Fonctions pour des expressions rationnelles compatibles Perl

Fonction	Description
`array preg_grep ( string motif, array tableau)`	Retourne dans un tableau les éléments extraits d'un autre tableau qui correspondent au motif.
`int preg_match ( string motif, string chaîne [, array tableau])`	Retourne Vrai si la chaîne contient le motif donné et emplit le tableau éventuel avec les correspondances.
`int preg_match_all ( string motif, string chaîne [, array tableau])`	Renvoie dans un tableau toutes les occurrences du motif dans la chaîne. Un ordre peut être envoyé par le paramètre optionnel.
`mixed preg_replace ( mixed motif, mixed nouveau_motif, mixed chaîne [, int limite])`	Remplace le motif par le nouveau motif dans la chaîne et retourne la nouvelle chaîne. Si la limite est posée, le nombre de remplacements sera limité.
`array preg_split ( string motif, string chaîne [, int limite])`	Découpe une chaîne en utilisant le motif comme séparateur.

Un certain nombre d'options affinent encore la recherche. Ces expressions rationnelles compatibles avec le langage Perl sont caractérisées par des contenants qui sont généralement des barres obliques ou peuvent être # ou %… Tout caractère non utilisé dans le motif recherché peut être mis à contribution. Reprenons l'exemple de l'extraction du nom du fichier dans le mini-forum :

```
preg_match("#^/.*/(.*)\.[A-z0-9]{3,4}$#", $_SERVER['PHP_SELF'], $tableau);
```

Ici, les caractères de limite d'expression sont #. Le premier caractère ^ désigne un début de ligne alors qu'il signifierait une négation dans des crochets. Le dernier caractère $ signifie une fin de ligne. Nous avons utilisé # et non /, déjà utilisé dans le motif recherché : un chemin avec des répertoires.

Ensuite, le premier caractère est une barre oblique présente au début de la valeur de `$_SERVER['PHP_SELF']` qui affiche le chemin à partir du répertoire web. Nous avons `.*`. Le point représente n'importe quel caractère et n'importe quel nombre de caractères, il peut s'agir de plusieurs répertoires ou d'aucun. En suivant l'expression, nous trouvons une autre barre oblique. Si nous regardons plus loin, nous trouvons le schéma du fichier avec le point et les trois ou quatre caractères qui caractérisent son format appelés son extension (*.doc*, *.jpg*, *.php*, *.html*, *.php3*…). Ce que nous voulons récupérer est le mot entre la dernière barre oblique et le point. Nous caractérisons l'extension, le point et la barre oblique pour délimiter le motif qui représente notre objectif. Cet objectif est représenté par `(.*)`.

Tab. 8.8 : Les caractères d'échappement dans les expressions rationnelles

Caractère	Description
`\w`	Caractère de mot. Les caractères de mot sont les alphanumériques et le blanc souligné (_).
`\W`	Caractère de non mot
`\b`	Limite de mot (entre le `\W` et le `\w`)
`\B`	Limite de non mot
`\d`	Caractère numérique
`\D`	Caractère non numérique
`\n`	Caractère de nouvelle ligne
`\s`	Caractère d'espace
`\S`	Tout caractère sauf un caractère d'espace

Créons un système de fichier très simple. Imaginons que vous ayez un répertoire où vous voulez placer tous vos fichiers sans fioritures. Ce qui vous intéresse et intéresse vos visiteurs, c'est seulement le contenu des fichiers. Vous avez peu de temps et acceptez de placer votre texte dans un modèle ou gabarit mais il vous pèse d'ajouter les liens de navigation à chaque nouveau fichier. Nous allons donc créer un fichier qui sera à l'écoute de votre répertoire. Dans un premier temps, ce fichier ira chercher la liste des fichiers dans une table. Chaque fichier sera affiché selon ses

caractéristiques : nom, titre, date de création, et dans l'ordre de date, c'est-à-dire que les plus récents seront en haut de la liste.

Il subsiste une objection, car certains fichiers seront pérennes et d'autres circonstanciels. Pour différencier les deux types de fichier, nous appellerons les fichiers d'actualité par un nom de date de type AAAAMMJJ. Plutôt que de mettre la date du jour, nous mettrons la date de péremption. Ainsi il sera simple de les archiver dès que la date sera dépassée.

Rentrons dans un tableau les types de fichiers qui seront affichés. Ainsi, il est inutile d'afficher les fichiers *.gif* ou *.jpg*.

```
<?
include_once "../commun/fonctions.inc.php";
include_once RACINE."/commun/connexion.inc.php";
$extensions=array ('htm', 'html', 'php', 'php3');
/*si $archi est égale à 0 c'est un fichier normal, si c'est 1,
 cette application liste les fichiers archivés. Ce sera le même
 fichier enregistré sous un autre nom comme archives.php*/
$archi='0';
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
 <title>La page web</title>
</head>
<body bgcolor="#FFFFFF" text="#000000" link="#0033FF" vlink="#990000"
 alink="#FF0000">
<table width="90%" border="0" cellspacing="2" cellpadding="2">
  <tr>
    <td colspan="2">
      <div align="center"><b>
<font face="Arial, Helvetica, sans-serif" size="3">
        LES PAGES WEB</font></b></div>
    </td>
  </tr>
   <tr>
    <td colspan="2" align="left">
<table width="50%">
<?
/*Si vous mettez le fichier dans le répertoire, laissez ces 3 lignes
 sinon utilisez la variable $rep en dessous qui est inactive (mise
 sous commentaire)*/
preg_match("#^(.*)/(.*)(\.)([A-z0-9]{3,4})$#",
 $SCRIPT_FILENAME, $tableau);
$rep=$tableau[1];
/*nous stockons le nom du fichier pour qu'il ne soit pas mis
```

```
 dans la base*/
$fichier_courant=$tableau[2].$tableau[3].$tableau[4];
//$rep = "../BonsVivants";
///AFFICHAGE DES DONNÉES//////
/*ici nous changeons la valeur selon que le fichier soit destiné
 à des archives ou à des fichiers pérennes*/
$sql="select * from fichiers where archive='$archi' OR archive=''
 ORDER BY moment DESC";
$resultat = @mysql_query($sql, $id_link);
$nombre = @mysql_num_rows($resultat);
if ($nombre>0){
while($rang=mysql_fetch_array($resultat)){
$nom_fichier=$rang['nom_fichier'];
$titre=$rang['titre'];
$archive=$rang['archive'];
$moment=$rang['moment'];
$moment_crea=date("j/n/Y",$moment);
echo "<tr><td><A HREF=\"$nom_fichier\" target=\"_blank\">$titre</a>
</td><td><i>$moment_crea</i></td></tr>\n";
}
}
else {
echo "<td><td colspan=2>Aucun fichier n'est pour l'instant disponible.
 Revenez bientôt sur cette page.</td></tr>";
}
///fin de l'affichage/////////////
////fourniture de la base de données//////
if (!$archi){
$aujourdhui=date("Ymd");
//cherchons la date la plus récente///
$sql="select MAX(moment) as date_limite from fichiers";
$resultat = @mysql_query($sql, $id_link);
$nombre = @mysql_num_rows($resultat);
if ($nombre==1){
//la date dans la base est entrée en TIMESTAMP UNIX//////
```

TIMESTAMP

Le TIMESTAMP Unix est le nombre de secondes depuis le 1er janvier 1970 alors que le TIMESTAMP MySQL est un nombre de type AAAAMMJJHHMMSS.

```
$rang=mysql_fetch_array($resultat);
$date_limite=$rang['date_limite'];
$maintenant=date("Ymd",$date_limite);
```

```
}
else {
$date_limite=time()-(60*60*24*365*20);
$sql="CREATE TABLE fichiers (
  clef int(11) NOT NULL auto_increment,
  nom_fichier varchar(60) NOT NULL default '',
  titre varchar(100) NOT NULL default '',
  archive char(1) NOT NULL default '1',
  moment bigint(20) NOT NULL default '0',
  PRIMARY KEY  (clef),
  UNIQUE KEY clef (clef),
  KEY clef_2 (clef)
)";
@mysql_query($sql, $id_link);
}
```

Passons en revue maintenant tous les fichiers du répertoire. Un test vérifie que le fichier n'est pas un répertoire ou ce fichier-ci :

```
$dir = opendir($rep);
while ($f = readdir($dir)) {
if(!is_dir($f) && $f!=$fichier_courant){
/*CHERCHER LE TITRE DU FICHIER DANS LES BALISES HTML
 <TITRE> ET </TITRE>*/
$fd_txt=fopen($f,'r');
$contenu_txt = fread ($fd_txt, filesize ($f));
fclose($fd_txt);
preg_match ("/(<(title)[^>]*>)+?((.|\n)*?)(<\/\\2>)/i",
 $contenu_txt, $trouvailles);
/*le modificateur i après le dernier délimiteur / dans
 l'expression permet de rendre le motif insensible à la casse*/
$titre=$trouvailles[3];
$titre=ereg_replace("\n", "", $titre);
```

Le moment a été mis sous cette forme dans le fichier : <!-- <date>20/02/2003</date> ->. Notez qu'il est en commentaire dans le HTML, donc non affiché. Si ce code existe dans le fichier, cette date sera prise en compte, sinon ce sera la date de dernière modification du fichier.

```
preg_match ("/(<(date)[^>]*>)+?((.|\n)*?)(<\/\\2>)/i",
 $contenu_txt, $trouvailles);
$ce_moment=$trouvailles[3];
if (strlen($ce_moment)){
/* ce moment a été mis sous cette forme dans le fichier
 <date>20/02/2003</date> */
$moments=split('/',$ce_moment);
```

```
$moment=mktime(0, 0, 0, $moments[1], $moments[0], $moments[2]);
}
////////////////////////////////////////
//donne la date du dernier changement dans le fichier//
$moment=filectime($f);
$temps_fichier=date("Ymd",$moment);
preg_match("#^(.*)\.([A-z0-9]{3,4})$#", $f, $tableau);
$nom=$tableau[1];
$extension=$tableau[2];
```

Comparons la dernière date où cette application a opéré pour entrer les fichiers dans la base et celle du fichier. Si celle du fichier est plus récente, ses caractéristiques sont insérées dans la base.

```
if (ereg("[0-9]{8}",$nom) && $nom>$aujourdhui && $moment>$date_limite
 && in_array ($extension, $extensions)){
$sql="insert into fichiers (nom_fichier, titre, archive, moment)
 VALUES ('$f', '$titre', '0', '$moment')";
@mysql_query($sql, $id_link);
}
elseif (in_array ($extension, $extensions) && $moment>$date_limite){
$sql="insert into fichiers (nom_fichier, titre, archive, moment)
 VALUES ('$f', '$titre', '0', '$moment')";
@mysql_query($sql, $id_link);
}
}
 }
closedir($dir);
}
///fin archi ==0
?>
</table></td></tr>
</table>
```

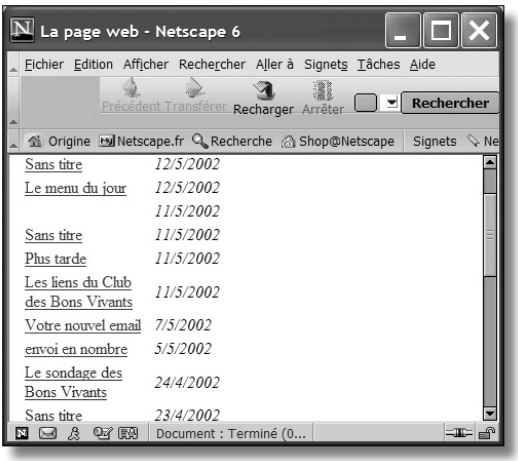
Si le fichier possède un nom de type *20020601*, ce nom sera comparé à la date du jour. S'il est moins récent, la valeur d'archive sera mise sur 1. Ceci est intéressant pour des fichiers d'actualité qui deviendront obsolètes après une certaine date et que nous archivons.

Listing 8.2 : *Ecoute fichiers : table_ecoute.php*

```
<?
if (!$archi){
$sql="select * from fichiers where archive='0' OR archive=''";
$resultat = @mysql_query($sql, $id_link);
while($rang=mysql_fetch_array($resultat)){
$nom_fichier=$rang['nom_fichier'];
```

```
$clef==$rang['clef'];
preg_match("#^(.*)\.([A-z0-9]{3,4})$#", $nom_fichier, $tableau);
$nom=$tableau[1];
if (ereg("[0-9]{8}",$nom) && $nom<$aujourdhui){
$sql="update fichiers set archive='1' where
 nom_fichier='$nom_fichier'";
@mysql_query($sql, $id_link);
}
}
}
?>
<p> </p>
</body>
</html>
```

Nous allons chercher la date de dernier changement. Si vous vous contentez de changer votre fichier de répertoire, cette date ne sera pas modifiée. Pour la modifier, il faut faire un changement dans le fichier. Si une fois que vous l'avez entré, vous modifiez ce fichier, il apparaîtra deux fois dans la liste avec deux dates différentes. Avant de faire cette manipulation, enlevez son entrée dans la base ou mettez un fichier plus récent dans le répertoire juste avant, cela permettra de repousser la date limite. Votre fichier plus ancien passera le contrôle sans problème.



▲ Figure 8.3 : *Une table d'écoute : attention de mettre un titre pour chaque fichier*

Nous préférons afficher une autre date pour des raisons d'actualité. Nous récupérons la date que nous avons ajoutée dans le fichier lui-même en l'encadrant de balises de notre invention : `<date>21/09/1999</date>`. Nous prenons soin de l'insérer dans un marqueur de commentaire HTML pour qu'il ne soit pas affiché. Ici, l'expression rationnelle joue le rôle d'un analyseur XML. L'expression rationnelle utilisée par les fonctions compatibles Perl utilise des délimiteurs et des modificateurs. Ici, le modificateur `i` rend l'expression insensible à la casse.

Les motifs sur plusieurs lignes

Le caractère point représente n'importe quel caractère, sauf les caractères de nouvelle ligne `\n`. Pour que ce caractère soit assimilé au point, il faut utiliser le modificateur `s`.

Ce système de fichiers peut être complexifié à l'échelle d'un site, en prenant soin de signaler les répertoires privés ou les répertoires destinés à classer les images du site pour les protéger. Vous pouvez aussi les afficher, comme nous l'avons fait pour l'annuaire de liens.

Tab. 8.9 : Modificateurs

Modificateur	Description
`A`	Cherche le motif seulement au début de la cible.
`E`	Cherche le motif seulement à la fin de la cible. Le caractère est pris comme caractère.
`U`	S'arrête dès que le premier motif qui satisfait à la recherche est trouvé.
`i`	Le motif est insensible à la casse.
`m`	Permet d'utiliser les ancrages ^ et $ pour des débuts ou des fin de ligne.
`s`	Force le moteur à considérer le caractère `\n` comme assimilé dans le caractère "point".
`x`	Permet de placer des espaces dans votre expression dont le moteur ne tiendra pas compte. Sert à rendre vos expressions plus lisibles.

Nous avons vu ce qui concernait les quantificateurs dans la partie POSIX. Il est possible avec les expressions rationnelles compatibles Perl d'utiliser des quantificateurs non avides. Cela signifie qu'ils ne complètent pas la recherche du motif une fois que le motif satisfait les critères. Ils ne s'arrêtent pas au premier motif mais ils n'essaient pas non plus à compléter indéfiniment le motif trouvé par d'autres caractères possibles.

Tab. 8.10 : Quantificateurs pour une recherche minimale

Non avide	Nombre de fois
`*?`	0, 1 ou plus
`+?`	1 ou plus
`??`	0 ou 1
`{min, max}?`	Au minimum min et au maximum max
`{min, }?`	Au moins min
`{nombre}?`	Exactement nombre

Quelques études de cas

Au cours de ces pages, nous avons utilisé des expressions rationnelles compatibles Perl sans les expliquer. Reprenons-les :

```
preg_match("#^(.*)\.([A-z0-9]{3,4})$#", $f, $tableau);
$nom=$tableau[1];
$extension=$tableau[2];
```

Dans cette expression, les délimiteurs sont #. Nous recherchons un motif qui tient sur une ligne, ce qui simplifie la recherche car cela évite l'avidité du moteur. Sur des expressions avides, le moteur ne se contente pas de trouver le motif, il a la tentation d'aller le chercher toujours plus loin. Le point précédé d'un signe d'échappement est un vrai point et non un caractère joker. Ensuite, l'extension du fichier peut contenir des lettres mais aussi des chiffres, au nombre de trois ou quatre.

```
$heure=preg_replace("/\d{8}(\d{2})(\d{2})\d{2}/","\\1",$datedujour);
```

Cet exemple prend une date de type AAAAMMJJHHMMSS dont on extrait l'heure, c'est-à-dire la première parenthèse. Le motif est composé de huit chiffres suivis d'une série de trois fois deux chiffres.

L'atome

L'atome est la plus petite particule de la matière. En grec, atomos signifie ce qui ne peut être divisé.

```
$contenu_txt=@preg_replace("#(\s)((\S)+?(@)+?(\S)+?(\.)+?
[A-z0-9_-]{2,3})(((\.)*<br>\s)|((\.)*
<p>\s)|((\.)*</p>\s)|((\.)*\s))#i",
 "\\1<A HREF=\"mailto:\\2\">\\2</a>\\7", $contenu_txt);
```

Ce motif est celui d'une adresse e-mail à laquelle nous ajoutons un lien mailto. La seule obligation concernant cette adresse est qu'elle soit encadrée d'espaces ou de balises de fin de ligne. Décomposons maintenant cette expression dans un tableau.

Tab. 8.11 : Analyse de la première partie de l'expression : ("#(\s)((\S)+?(@)+?(\S)+?(\.)+?[A-z0-9_-]{2,3})

Atome	Description
(\s)	Espace
(\S)+?	Au moins un non espace (un caractère différent d'un espace)
(@)+?	Au moins un arobace
(\S)+?	Au moins un non espace. Juste après l'arobace, le moteur continuera sa route aussi loin qu'il ne rencontrera pas d'espace et jusqu'au point qui suit.
(\.)+?	Au moins un point
[A-z0-9_-]{2,3})	Après le point entre deux et trois caractères alphanumériques ou blanc souligné ou tiret

**Tab. 8.12 : Analyse de la seconde partie de l'expression : (((\.)*
\s)|((\.)* \s)|((\.)* \s)|((\.)*\s))**

Atome	Description
((\.)* \s)	Point éventuel suivi d'une balise et un espace comme une fin de ligne
((\.)* \s)	Idem avec la balise
((\.)* \s)	Idem avec la balise
((\.)*\s)	Idem sans balise

Ces atomes sont reliés par le signe | (ou). Ainsi, les signes qui suivent la partie intéressante, l'adresse e-mail, sont une fin de ligne ou un espace blanc.

Nous pouvons améliorer l'expression en enlevant la deuxième ligne de cette deuxième partie et en modifiant la ligne suivante :

```
((\.)*</*p>\s)
```

Ainsi, la ligne oblique peut être ou non présente.

À la fin du premier membre de l'expression qui consiste en ces deux parties, nous avons ajouté le modificateur `i` après le délimiteur pour que l'expression soit insensible à la casse.

```
"\\1<A HREF=\"mailto:\\2\">\\2</a>\\7"
```

Nous utilisons dans le deuxième membre de l'expression, la récursivité avec les deux barres obliques inversée suivies d'un chiffre qui correspond à la place des parenthèses de la gauche vers la droite et du général au particulier. La parenthèse 0 est l'ensemble du motif, la parenthèse 1 est ici un espace et la 2, l'adresse e-mail. Nous terminons par la septième parenthèse qui est l'espace ou la fin de ligne terminant le motif :

```
preg_match ("/(<(title)[^>]*>)+?((.|\n)*?)(<\/\\2>)/i",
 $contenu_txt, $trouvailles);
$titre.= " ".$trouvailles[3];
```

Ici, nous cherchons à capturer le contenu des balises <title> et </title>, en incluant l'éventualité qu'elles puissent chevaucher une ligne. L'atome que vous rencontrez au milieu `((.|\n)*?)` est ce que l'on cherche vraiment, c'est-à-dire tous les caractères compris entre les deux balises.

Le caractère point

Le caractère point ne comprend pas le caractère de nouvelle ligne.

Le premier atome est la première balise. Il est composé du premier chevron suivi de `title` puis d'un motif assez obscur : `[^>]*`, présent pour éviter que le moteur soit trop gourmand mais qu'il exige malgré tout la présence du chevron fermant. C'est le genre d'expression que l'on trouve à la suite de tests et d'approximations. Le motif finit avec la répétition de `title` (`\\2`) encadré par des chevrons. Le tout est modifié par `i` qui le rend insensible à la casse. Être un orfèvre en expressions rationnelles demande de l'intuition et de l'expérience. Ce n'est pas une science exacte mais un outil indispensable aux possibilités encore largement inexplorées.

8.3 Conclusion

Les expressions rationnelles demandent beaucoup de pratique car chaque problème posé est unique. La construction d'une expression part d'une base de caractères puis, par approximations, cherche à atteindre la précision requise. Cette précision se nomme la granularité. L'objectif étant d'atteindre la granularité la plus appropriée. Si elle est trop fine, vous ne trouverez rien. Si elle est trop grossière, vous aurez trop de résultats. Vous le voyez dans les moteurs de recherche. En étant trop précis, vous n'obtenez aucun résultat.

9

Parsez-vous XML ?

Le XML est arrivé à point pour répondre à des besoins de communication entre les applications avec un langage format texte. Il répond, en outre, à ce vieux rêve de clairement séparer le contenu de la présentation. Avec le XML, la présentation et le contenu deviennent réutilisables dans différents contextes (Web, archives, flux, téléphonie mobile...).

L'objet de ce livre est non pas de vous donner une étude exhaustive du XML en général mais de vous permettre de créer des applications utiles ; vous pourrez ainsi approfondir le sujet avec une bonne base de compréhension.

PHP dispose de modules dédiés au XML avec les fonctions adéquates pour répondre à toutes les circonstances.

Avant de créer vos pages XML sur votre serveur web…

Vérifiez avec la fonction `phpinfo()` si vous disposez du XML et du module Sablotron pour le XSLT.

9.1 Un peu d'histoire...

Le XML est un langage de balises comme le HTML. Ces deux langages sont issus d'un ancêtre commun, le SGML. Le SGML était trop complexe pour se répandre, alors est né le HTML, qui en est une simplification et qui caractérise aussi bien une forme (<font>, <p>, <b>...) qu'un contenu (<title><script>...), utilisant la navigation par hyperliens. Le HTML a fait exploser la production de pages sur l'Internet grâce à sa simplicité et à la tolérance des navigateurs quant à la grammaire des balises pour la restitution de ces pages. Face à la prolifération des informations, voire à leur saturation, il a fallu trouver une façon de caractériser leur contenu pour une communication plus pointue, et c'est là qu'intervient le XML. La plupart du temps, l'application XML se compose de trois fichiers : le fichier DTD (`.dtd`), une feuille XSLT(`.xsl`) et le fichier XML proprement dit (`.xml`).

9.2 Le fichier XML

Le fichier XML est composé de deux parties complémentaires.

L'espace de nommage

Vous commencez par déclarer la version XML utilisée pour stocker les données ainsi que le type d'encodage :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Ce type d'encodage vous permet de rendre correctement les caractères propres à l'Europe occidentale.

Sur la ligne en dessous, vous déclarez la DTD :

```
<!DOCTYPE nomDTD SYSTEM "nom.dtd">
```

Vous nommez la DTD et indiquez le chemin vers le fichier ou l'URL. Ensuite, vous déclarez le fichier XSLT, qui mettra le fichier XML en forme :

```
<?xml-stylesheet type="text/xsl" href="adherents.xsl"?>
```

Les éléments et les attributs

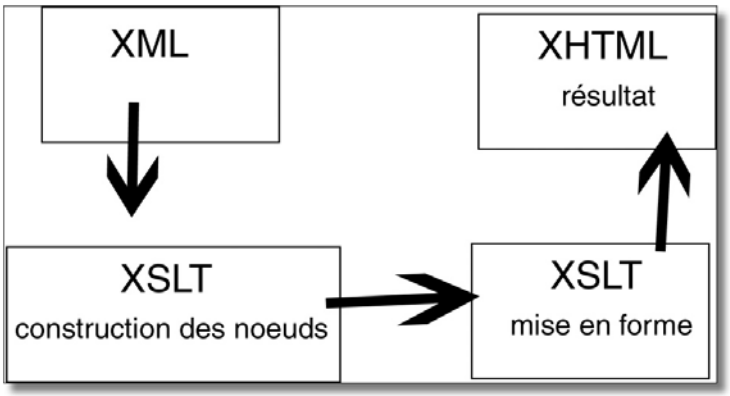
Le fichier XML est formé de balises comme l'exemple ci-dessous :

```
<fiche>
<img src="jeandupond.jpg"/>
<prênom>Jean<prênom>
<nom>Dupont</nom>
<adresse>16, rue du Pont Louis-Philippe</adresse>
<codepostal>75004</codepostal>
<ville>Paris</ville>
</fiche>
```

Chaque balise ouvrante renvoie à une balise fermante rigoureusement positionnée pour encadrer symétriquement des données, excepté la balise `img`, qui se termine ici par / du fait que c'est un élément isolé. Les balises sont nommées des éléments tandis que leur ou leurs propriétés comme la

propriété `src` de l'élément `img` sont appelées des attributs. Ce que contient l'élément est du texte.

Le fichier XML doit avoir la bonne forme. Si vous choisissez des capitales pour nommer les balises, vous devrez vous y tenir, qu'elles soient ouvrantes ou fermantes. Vous allez donc créer un fichier de DTD dans lequel vous consignerez les règles que vous appliquerez à vos balises. Il servira à valider votre fichier et à repérer les erreurs de syntaxe. Vous pourriez ainsi décider que telle balise est obligatoirement à l'intérieur de telle autre balise...



▲ Figure 9.1 : *Processus XSLT*

Un fichier XML est un fichier brut de données, sans aucune présentation mais ponctué de balises ; il sera donc accompagné d'un fichier XSLT. Ce fichier de présentation est parcouru et interprété par une application serveur qui va fusionner les fichiers pour fabriquer une page résultat qui sera soit un vrai fichier, soit l'écran par défaut. Le langage XSLT est déconcertant pour tout programmeur formé aux langages procéduraux ou à objets. Il est fondé sur la création de règles de modèle (*template*) à partir du fichier source. C'est sur l'arbre résultat que les transformations se font pour présenter les données d'une manière lisible et agréable pour tout un chacun.

Le fichier XML est traité (*parsed*) par un analyseur XML qui est par défaut le module *Expat*. Ce fichier correspond au contenu de la page. C'est

ce document qui sera modifié dans le cadre d'une mise à jour. Comme les balises sont explicites et le contenu, écrit sans codage, n'importe quel utilisateur à qui on a enseigné deux ou trois règles saura modifier ce type de fichier sans dommages.

Tab. 9.1 : Codage des caractères dans un fichier XML

Caractère	Codage	Statut
&	`&`	Obligatoire
<	`<`	Obligatoire
>	`>`	Obligatoire
'	`'`	À l'intérieur d'un attribut délimité par des apostrophes
"	`"`	À l'intérieur d'un attribut délimité par des guillemets

Quelques règles :

- Les balises doivent être rigoureusement de la même casse, généralement en minuscules.
- Toute balise ouvrante doit être accompagnée par une balise fermante.
- Le nom d'une balise ou d'un attribut doit débuter par une lettre ou un souligné _ suivi éventuellement par plusieurs lettres, chiffres, points, tirets ou soulignés.
- Une balise sans contenu sera finie par />, par exemple `<br/>` ou `<br />`.
- Les attributs sont entre guillemets ou entre apostrophes, par exemple `<page numéro="1">`.
- Les balises encadrant différentes entités ne peuvent se chevaucher, elles doivent s'emboîter ou se suivre.
- Les caractères guillemet sont codés quand ils appartiennent à un attribut délimité par des guillemets.
- Les caractères apostrophe sont codés quand ils appartiennent à un attribut délimité par des apostrophes.
- Les noms de balise ou d'attribut ne peuvent contenir d'espace.

- Les balises ou les attributs ne peuvent avoir un nom qui commence par `xml` ou `XML` ou `Xml`…

Comme auteur d'un fichier XML, vous nommez vous-même les différentes balises afin que le sens du contenu soit explicite pour un utilisateur :

```
<titre>Php &amp;amp; MySQL en ligne<sous-titre>L'essentiel pour concevoir
un site web dynamique!</sous-titre><collection
référence="3679">e-poche</collection></titre>
```

Ce que nous venons d'énoncer est essentiel pour que le document XML soit dans une bonne forme ; le document, appelé la DTD, qui accompagne ce fichier permettra de valider.

9.3 La DTD

La DTD est un document qui édicte les règles des entités composant le fichier XML, c'est-à-dire que vous allez décrire la succession et l'imbrication des éléments et des attributs.

Les différents types de déclarations

Les différents éléments et attributs sont déclarés en décrivant leur contenu.

Tab. 9.2 : Différentes déclarations dans une DTD

Déclaration	Explication
<!ELEMENT titre (#PCDATA)>	L'élément `titre` contient du texte traité.
<!ELEMENT titre (sous-titre, collection)>	L'élément `titre` contient les éléments `sous-titre` et `collection`.
<!ELEMENT titre (sous-titre, collection+)>	L'élément `titre` contient l'élément `sous-titre` et une à plusieurs balises `collection`.
<!ATTLIST collection référence CDATA>	L'élément `collection` a comme attribut `référence` qui est de type CDATA (texte)

Prenons l'exemple d'un fichier XML tiré du site www.accesites.com :

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<?xml-stylesheet type="text/xsl" href="index.xsl"?>
<!DOCTYPE index SYSTEM "index.dtd">

<page>

<paragraphe><gras>Vous voulez un site sans surprise, tout compris,
réalisé dans de moindres délais, simple à mettre à jour en toute
indépendance ? </gras> </paragraphe>

<paragraphe><orange>Nos packs sont faits pour
vous.</orange></paragraphe>

<paragraphe> Bénéficiez de notre <gras>savoir-faire</gras> dans un
site à <gras>vos couleurs</gras>.</paragraphe>

<paragraphe> Vos clients et vos prospects feront une agréable visite
sur votre espace grâce à une navigation simple et
directe.</paragraphe> <paragraphe> Si vous désirez une présence
encore plus personnalisée nous pouvons concevoir votre site
<gras>"sur mesure"</gras> sur simple devis gratuit.</paragraphe>

<paragraphe>Quel que soit votre choix, un grand nombre
d'<gras>options</gras> et de services adaptés à vos besoins, sont
à votre disposition.</paragraphe>

</page>
```

Vous avez ici des balises imbriquées. Elles ne sont pas seulement imbriquées car, si les données paragraphe contiennent parfois les balises orange ou gras, elles contiennent aussi du texte (voir fig. 9.2).

Voici la DTD :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT page (paragraphe+)>
<!ELEMENT paragraphe (gras*,orange*)>
<!ELEMENT gras (#PCDATA)>
<!ELEMENT orange (#PCDATA)>
```

Nous voyons ici que l'élément page contient un ou plusieurs éléments paragraphe. L'élément paragraphe, lui, ne contient aucun élément gras et orange ou en contient plusieurs. Les éléments gras et orange sont des données traitées par le processeur XSLT.

▲ Figure 9.2 : *Accesites.com*

9.4 Le XSLT

Le XSLT est un langage de type XML qui utilise la syntaxe et la sémantique d'un autre langage nommé *XPath*. Tout cela forme un vrai langage de programmation qui permet de créer des processus de grande complexité.

L'espace de nommage

Il s'agit d'une sorte d'en-tête du fichier. Il contient le type d'encodage des caractères (ici les caractères d'Europe occidentale) et la version de XML (en ce moment, c'est la version 1.0).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
```

La déclaration `xmlns` (*XML name space*) a un attribut qui ressemble à une URL mais, en fait, c'est juste une convention qui désigne le type de nommage qui sera utilisé.

La feuille de style

Ensuite, la feuille de style est contenue entre les balises `<xsl:stylesheet…>` et `</xsl:stylesheet>`, et vous pouvez y déclarer les différents modèles en commençant par le modèle de la racine. Ce modèle représente la forme générale de votre page. Ainsi, vous y mettrez votre charte graphique : le logo, la barre de menus… tout ce qui enveloppe les données. Vous partez d'un fichier HTML que vous normalisez en XHTML. Vous convertissez alors les balises `<br>` en `<br />`… selon les règles énoncées plus haut. Ensuite, vous caractérisez le `html` comme étant du :

```
<html lang="fr" xml:lang="fr" xmlns="http://www.w3.org/1999/xhtml">
```

Le fichier HTML original peut contenir des caractères que vous ne voulez pas soumettre au traitement XSLT, comme votre code javascript dans l'en-tête. Pour cela, vous l'introduisez dans la balise `<![CDATA[…]]>` :

```
<script language="JavaScript"><xsl:comment><![CDATA[
function MM_preloadImages() { //v3.0
   var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
   var i,j=d.MM_p.length,a=MM_preloadImages.arguments;
   for(i=0; i<a.length; i++)
      if (a[i].indexOf("#")!=0){
         d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
]]></xsl:comment>
</script>
```

Cela ne vous empêche pas d'utiliser aussi des feuilles de style CSS dans votre document résultat.

```
<link rel="stylesheet" href="accesites.css" type="text/css"/>
```

Les modèles

Votre modèle (*template*) est encadré par les balises <xsl:template> et </xsl:template>. L'ouverture de la balise utilise l'attribut match, qui identifie l'élément qui correspond au modèle dans le fichier XML :

```
<xsl:template match="/">
```

Ici, le signe barre oblique désigne l'élément racine.

Vous introduisez votre code entre vos balises. À l'endroit où vous devez introduire des données xml qui sont sous l'arborescence du modèle courant, vous appelez le modèle qui leur correspond (match) soit par son nom de balise, soit par un terme générique. Voici l'exemple expurgé de nombreuses lignes de code javascript et html (ne le prenez pas littéralement), car c'est ici le code XSLT qui nous intéresse. Dans le fichier complet, l'appel au modèle de l'élément paragraphe est inséré dans un tableau HTML.

```
 <xsl:template match="/">
<html lang="fr" xml:lang="fr" xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>AcceSites</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1"/>

<script language="JavaScript"><xsl:comment><![CDATA[
function MM_preloadImages() { //v3.0
   var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
      var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0;
      i<a.length; i++)
      if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image;
      d.MM_p[j++].src=a[i];}}
}
]]></xsl:comment>
</script>

<link rel="stylesheet" href="accessites.css" type="text/css"/>
</head>
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0" topmargin="0"
marginwidth="0" marginheight="0"
onLoad="MM_preloadImages('images/boutpack_over.gif','images
/boutoption_over.gif','images/boutcommand_over.gif','images
/boutcontact_over.gif','images/boutmarche_on.gif','images
/boutdevis_on.gif')">
```

```
<xsl:apply-templates select="page/paragraphe"/>

</body>
</html>
</xsl:template>
```

Ensuite, nous ajoutons les modèles des différents éléments :

```
<xsl:template match="paragraphe">
   <p class="courant">
<xsl:apply-templates/>
</p>
   </xsl:template>
```

Il s'agit ici d'un paragraphe ; nous appliquons donc les balises et . <xsl:apply-templates/> est une commande générique qui évite de nommer chaque modèle à appeler et qui appelle donc le contenu du paragraphe dont les contenus gras et orange en vérifiant leur modèle.

```
   <xsl:template match="gras">
<b><xsl:value-of select="."/></b>
   </xsl:template>
   <xsl:template match="orange">
   <p class="sstitre"><xsl:value-of select="."/></p>
   </xsl:template>
```

Nous voyons apparaître ici la commande <xsl:value-of select="."/>, qui représente la valeur de l'élément courant. On aurait pu aussi nommer chaque élément.

Comme vous pouvez le constater, nous pouvons envisager la structure de données d'un fichier XML comme une arborescence ou une imbrication de données de type poupées russes. C'est un langage déclaratif fondé sur des modèles (templates).

Notre ouvrage n'est pas un livre sur le langage XML, et il s'agit donc là d'une introduction pour vous aider à aborder ce langage. Dans ce chapitre, nous ne faisons qu'effleurer le XML avec des exemples pratiques. Il reste à étudier maintenant le fichier PHP qui appelle les processus de traitement de ces fichiers.

9.5 Le fichier PHP

Le fichier PHP appelle les différents fichiers qu'il envoie vers les API pour qu'ils soient traités en vue d'un fichier résultat. Voici un exemple de code :

```
$xslt=xslt_create();
$resultat=xslt_process($xslt, 'index.xml', 'index.xsl');
echo $resultat;
xslt_free($xslt);
```

Le programme crée un analyseur XSLT par la fonction `xslt_create()` puis il crée avec l'analyseur, le fichier XML et le fichier XSLT un fichier résultat qui sera ici affiché à l'écran avant d'être effacé de la mémoire du serveur. Cette fonction a trois arguments supplémentaires optionnels qui permettent de passer des paramètres au processeur (ici *Sablotron*).

Le premier argument est le nom du fichier résultat, si vous voulez conserver ce résultat dans un fichier comme nous le verrons dans les prochains exemples. Si vous ne l'utilisez pas mais que vous utilisiez les arguments suivants, il vous faudra mettre `NULL` pour signaler son absence.

Le deuxième argument est un tableau d'arguments qui pour l'instant est réservé aux versions prochaines de *Sablotron*.

Le troisième est un tableau de paramètres au processeur XSLT, tels des paramètres de pagination.

Avant de lancer l'analyseur XSLT, le fichier php peut procéder à des opérations afin de les envoyer au processeur. Tous ces fichiers se marient harmonieusement en se complétant. PHP apporte sa richesse de fonctions et sa puissance, XML structure les données et les rend simples à mettre à jour, y compris par des non-spécialistes, XSLT ajoute une grande versatilité de format du même document (fichier HTML, fichier CSV…).

9.6 Magasin électronique en XML

La plupart du temps, une boutique électronique est une suite de pages qui affichent des articles avec leur nom, leur référence, leurs caractéristiques et leur prix accompagné d'un bouton ou d'un lien pour le stocker dans le caddie. Ce caddie n'est que la métaphore d'un tableau conservé en

mémoire, au fil de la navigation, sur un fichier, une base de données ou une variable de session.

Le côté fastidieux de ce genre d'application est l'élaboration du format des pages et de leur présentation, qu'il s'agisse de la boutique elle-même ou du contenu du caddie et enfin de la facturation. Le XSLT permet de créer un format une fois pour toutes puis de le réutiliser avec des modifications minimes. Dans ce chapitre, nous allons non pas explorer tous les aspects de cette problématique mais aborder la pagination et la manière dont vous pouvez concevoir votre application.

La conception de la boutique

Imaginons que vous ayez un client qui vous demande une boutique. Une boutique est non pas un monument mais un flux ; les articles changent comme les prix ou les modèles, il faut donc qu'il soit facile au commerçant de mettre à jour sa boutique sans avoir à faire appel à chaque fois au programmeur. Avec le XML, il pourra modifier directement le fichier XML. Ce dernier sera construit à partir de la base de données MySQL, qui elle-même sera nourrie par un fichier CSV créé à partir d'un tableau. Ainsi, le commerçant pourra faire ses modifications sur son tableur et modifier le contenu de son site.

Création du fichier XML

Le fichier original sera un fichier CSV, c'est-à-dire un fichier au format texte. Vous allez en récupérer les données une fois votre table créée à l'identique du fichier. Les lignes sont marquées par des retours à la ligne et les données sont séparées par un point-virgule. Veillez à ce qu'il n'y ait aucun point-virgule dans les données.

Le commerçant met le fichier `.csv` dans un répertoire réservé sur le serveur par ftp ou par un formulaire de téléchargement de fichier. Le programme peut le récupérer comme suit :

```
$lignes=file("boutique.csv");
foreach ($lignes as $ligne){
$donnees=explode(';', $ligne);
$article=$donnees[0];
$reference=$donnees[1];
$caracteristique=$donnees[2];
```

```
$prix=$donnees[3];
$sql="TRUNCATE TABLE 'boutique'";
mysql_db_query($dbname,$sql,$id_link);
$sql="insert into boutique (article, reference, caracteristique,
prix) VALUES ('$article', '$reference', '$caracteristique',
'$prix')";
mysql_db_query($dbname,$sql,$id_link);
//pour les détails concernant les bases de données, référez-vous  au
chapitre MySQL en action
}
```

Le programme fait ici une opération dangereuse, c'est-à-dire qu'il vide la table avant de la remplir. Il serait plus sage de copier les données dans une autre table puis de faire l'opération de remplacement à la main ou alors, si vous disposez d'une table temporaire, de la créer et de faire un `commit` et un `rollback` pour changer la table en la nouvelle sans risque. C'est une procédure que vous avez parfois dans *MySQL*(structure mode *InnoDB*) et dans *SQLite* et qui permet de valider toute la procédure de transaction avant de l'enregistrer définitivement dans la base. Si la procédure est interrompue, le programme suspend le `commit` par un `rollback` et l'opération est annulée. Vous incluez dans le programme l'envoi d'un e-mail qui vous avertit de l'opération.

Une fois les données entrées, vous allez pouvoir créer votre fichier XML. Les balises seront créées par le nom de colonnes. Nous y ajoutons aussi des balises pour les pages avec un numéro pour attribut et, comme élément racine, nous utilisons le nom de la table.

Au début du programme, les lignes de l'espace de nommage sont créées :

```
$nom_fichier="boutique.xml"; $nom_table="boutique";$nbre_par_page=20;
$fp = fopen($nom_fichier,"w"); // création du fichier de sortie
fwrite($fp,"<?xml version='1.0' encoding='ISO-8859-1'?>\n");
// la racine est $tab
fwrite($fp,"<!DOCTYPE boutique SYSTEM \"boutique.dtd\">\n");
fwrite($fp,"<?xml-stylesheet type=\"text/xsl\" href=\"
boutique.xsl\"?>\n");
fwrite($fp,"<$nom_table>\n<ladate>$ladate</ladate>\n");
```

Ensuite, le programme extrait les données de la base :

```
$sql="select * from $nom_table ORDER BY nom";
$resultat=mysql_db_query($dbname,$sql,$id_link);
$nombre=mysql_num_rows($resultat);
```

```
$i=1;
while($rang = mysql_fetch_array($resultat)) {
if ($i==1){
fwrite($fp,"<page numéro=\"1\"> \n");
}
elseif ($i%$nbre_par_page==1){
fwrite($fp,"</page>\n");
fwrite($fp,"<page
numéro=\"".($i+($nbre_par_page-1))/$nbre_par_page."\">\n");
//ici on ajoute 19
}
fwrite($fp,"<fiche numéro=\"$i\">\n");
//le numéro de fiche est optionnel dans le programme qui suit
$ajout=True;
foreach($rang as $cle=>$valeur) {
   $ajout = !$ajout;
//vrai une fois sur deux
//(essayez d'enlever cette variable pour comprendre)
   $valeur=str_replace(array("&", '"', "'", "<", ">"), array('&amp;',
    '&quot;', '&apos;', '&lt;', '&gt;'), $valeur);
   if ($ajout) fwrite($fp,"<$cle>$valeur</$cle>\n");  //if
} //fin foreach

fwrite($fp,"</fiche>\n");
if ($i==$nombre){
   fwrite($fp,"</page>\n");
} //fin if

$i++;
} //fin while

fwrite($fp,"</$nom_table>");
fclose($fp); // fermeture du fichier

mysql_free_result($resultat);
mysql_close();
////////////////FIN CREATION FICHIER XML////////////////////
```

Avec ce fichier XML, nous allons maintenant créer un fichier CSV.

Création d'un nouveau fichier CSV à partir d'un fichier XSLT

Nous avons vu précédemment que le fichier CSV est un fichier dont chaque rang de la base de données est une ligne et dont chaque colonne est délimitée par un point-virgule. Nous allons donc créer un fichier résultat au format CSV à partir du fichier XML *boutique.xml* et une feuille

de style XSLT. Dans le programme, nous introduisons l'affichage des erreurs avec les fonctions `xslt_error()` et `xslt_errno()`. Le fichier résultat se nommera *boutique.csv*.

```
////////creation CSV//////////////////
$xslt=xslt_create();
$resultat=xslt_process($xslt, 'boutique.xml', 'csv.xsl', 'boutique.csv');
if (!$resultat) {
   echo "Désolé, boutique.xml n'a pu être transformé par boutique.xsl
   en ";
   echo « boutique.xml. La raison est » . xslt_error($xslt) . « et »;
   echo « le code d'erreur est » . xslt_errno($xslt);
}
xslt_free($xslt);
////////FIN creation CSV//////////////////
```

Le fichier XSLT consiste en une concaténation des éléments séparés par une virgule qui est terminée par une fin de ligne. Une nouvelle balise apparaît dans la feuille de style : `<xsl:output/>` ; cette balise envoie l'attribut `method="text"` à l'intention du format du fichier résultat :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="text" omit-xml-declaration='yes'
encoding="ISO-8859-1"/>
<xsl:template match="fiche">
<xsl:value-of select="concat(article, ';', reference, ';',
caracteristique, ';', prix, ';', '&#xD;')" />
   </xsl:template>
   </xsl:stylesheet>
```

Le caractère `
` est une fin de ligne. Vous pouvez vous envoyer ce fichier pour avoir une copie de votre base.

Le fichier XSLT de la boutique

Ce fichier commence par l'espace de nom. La balise `<xsl:param/>` est déclarée, elle initialise les paramètres qui sont entrés par le fichier php. Ici, ces paramètres désignent la pagination de notre boutique. La page courante est initialisée à 1. Cette valeur est donnée par défaut, c'est-à-dire qu'elle est prise en compte si aucune valeur n'a été envoyée au programme XSLT. Les autres paramètres donnent la valeur de page

précédente et de la dernière page. Le paramètre script donne le nom du fichier php qui lance l'analyseur :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
<xsl:output method="html" encoding="ISO-8859-1"/>
<xsl:param name="courante" select="1"/>
<xsl:param name="precedente"/>
<xsl:param name="script" select="boutique.php"/>
<xsl:template match="/">
<html lang="fr" xml:lang="fr" xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>La boutique</title>
</head>
<body bgcolor="White" text="Black" link="Blue" vlink="Navy"
alink="Red">
<div align="center">
  <table width="95%" border="0" cellspacing="10" cellpadding="2">
  <tr>
  <td width="30%" align="center" valign="middle">
    <a href="http://apaer.org/accueil.htm">
    <img src="../images/logos/logo-nouvelle-page-accueil.jpg"
         width="130" height="130" border="0"/></a></td>
  <td width="50%" align="center"><br /></td>
  <td width="20%" align="center">
  </td>
  </tr>
  <tr>
     <td valign="top" colspan="3"><p align="center">
     <xsl:apply-templates select="boutique"/>
     </p>
     </td>
     </tr>
   </table>
</div>
</body>
</html>
</xsl:template>
<xsl:template match="boutique">
<table width="95%" border="0" cellspacing="10" cellpadding="2">
<xsl:apply-templates
select="page[attribute::numéro=$courante]/fiche"/>
<xsl:call-template name="pagination"/>
</table>
</xsl:template>
<xsl:template match="fiche">
<xsl:variable name="reference" select="ref"/>
   <tr>
```

```
      <td><xsl:value-of select="article"/></td>
      <td><xsl:value-of select="reference"/></td>
      <td><xsl:value-of select="caracterisque"/></td>
      <td><xsl:value-of select="prix"/> &euro;</td>
      <td><form action="{$script}"  method="post">
          <input type="hidden" value="{$reference}"
                 name="insert"/>
          <input type="text" value="1" size="2"
                 name="quantite"/>
          <input type="submit" value="mettre dans le caddie"
                 class="bouton_admin"/>
      </form></td>
   </tr>
</xsl:template>

<xsl:template name="pagination">
   <tr>
<xsl:variable name="ultime_page" select="count(./page)"/>
   <xsl:choose>
      <xsl:when test="$courante&lt;=1">
      <!-- Sur la page 1 la page précédente n'apparaît pas -->
         <td>première</td>
         <td>precedente</td>
         </xsl:when>
      <xsl:otherwise>
        <td><a href="{$script}?courante=1"><b>première</b></a>
        </td>
      <td><a href="{$script}?courante={$courante   -1}">
      <b>précédente</b> </a> </td>
      </xsl:otherwise>
   </xsl:choose>
   <td>
   <xsl:text>Page </xsl:text>
   <xsl:value-of select="$courante"/>
   <xsl:text> sur </xsl:text>
   <xsl:value-of select="$ultime_page"/>
   </td>

   <xsl:choose>
      <xsl:when test="$courante=$ultime_page">
         <!-- Sur la dernière page, pas de page suivante -->
         <td class="pagination">prochaine</td>
         <td class="pagination">dernière</td>
         </xsl:when>
      <xsl:otherwise>
         <td><a href="{$script}?courante={$courante +1}">
             <b>prochaine</b> </a></td>
      <td><a href="{$script}?courante={$ultime_page}">
             <b>dernière </b> </a></td>
      </xsl:otherwise>
```

```
      </xsl:choose>
   </tr>
</xsl:template>
</xsl:stylesheet>
```

Ce programme introduit deux types de données : les paramètres et les variables.

Nous avons vu que les paramètres sont envoyés à l'analyseur XSLT par le programme PHP. Les variables sont nourries à l'intérieur de la feuille de style.

La première variable porte la valeur du nœud `ref`. Pourquoi ne pas avoir utilisé `<td><xsl:value-of select="reference"/></td>` dans l'URL comme un peu plus haut ? Tout simplement parce que le document XSLT est un document XML et que cette forme contrevenait aux règles du XML. C'est donc une variable qui permet de donner la valeur de la référence qui est envoyée dans le tableau qui sert de panier (voir le chariot électronique dans le chapitre *Des algorithmes, des outils et des fonctions*).

La deuxième variable évite d'avoir à mettre le numéro de la dernière page en paramètre dans la page PHP ; ainsi, ce programme demande un minimum de modifications pour s'adapter à toutes les situations. La fonction `count()` compte le nombre de nœuds de l'élément page.

Le fichier PHP lance l'analyseur enrichi des paramètres de page et de script.

Le fichier PHP de la boutique

Le fichier PHP envoie un certain nombre de paramètres dans l'analyseur XSLT. Il reçoit parfois certaines de ces données dans l'URL comme le numéro de la page courante, ou alors les données sont nourries par les possibilités de PHP :

```
$xslt=xslt_create();
if (!$courante){
$courante=1;
}
/* le fait que $courante soit dans le tableau, le paramètre par
défaut dans la feuille XSLT ne fonctionnera pas car $courante vide
est pris en compte. Pour que le paramètre par défaut soit
enclenché, il faudrait que le paramètre ne soit pas présent dans
le tableau */
```

```
$params=array(courante=>$courante, script=>$PHP_SELF);
$arg=array();
$resultat=xslt_process($xslt, 'boutique.xml', 'boutique.xsl', NULL,
$arg, $params);
echo $resultat;
if (!$resultat) {
   echo "Désolé, boutique.xml n'a pu être transformé par boutique.xsl
   en ";
   echo « boutique.xml. La raison est » . xslt_error($xslt) . « et »;
   echo « le code d'erreur est » . xslt_errno($xslt);
}
xslt_free($xslt);
```

Le XML enrichit considérablement le confort du PHP et l'élaboration d'une pagination pour une boutique, une page de résultat. Nous aurions pu ajouter bon nombre d'applications mais ce n'est pas l'objet de ce livre. Avec PHP 5 apparaît une nouvelle implémentation du XML, *SimpleXML*.

9.7 SimpleXML

Le XML peut être assez complexe, comme vous l'avez vu précédemment, en particulier lorsqu'il est accompagné du XSLT. L'application *SimpleXML*, qui est livrée avec PHP 5, court-circuite le XSLT. Le fichier XML est utilisé comme une base de données objet. Les données peuvent alors être insérées dans un fichier php.

Prenons un fichier XML très simple :

```
<?xml version="1.0"?>
<librairie>
<livre>
<titre> L'Etrusque </titre>
<auteur> Mika Waltari </auteur>
<prix>23,66</prix>
</livre>
<livre>
<titre>Les amants de Byzance </titre>
<auteur> Mika Waltari </auteur>
<prix>9,99</prix>
</livre>
</librairie>
```

Avec PHP 5, pour l'interpréter, il suffira de faire :

```
$population = simplexml_load_file('librairie.xml');

foreach($librairie->livre as $livre) {
   echo 'Titre : ' ,$livre ->titre.'<br>';
   echo 'Auteur : ' ,$livre ->auteur.'<br>';
   echo 'Prix : ' ,$livre ->prix.'&euro;<br><br>';
}
```

```
Titre : L'Etrusque

Auteur : Mika Waltari

Prix : 23,66€

Titre : Les amants de Byzance

Auteur : Mika Waltari

Prix : 9,99€
```

SimpleXML est rapide à mettre en place et pratique tant que vos applications ne sont pas trop complexes. Il vous permettra par exemple d'ajouter sur votre site des syndications de nouvelles en RSS (*Really Simple Syndication* ou *Rich Site Summary*).

Le RSS

RSS est un format de fichier XML standard qui comprend une balise `item` contenant les balises `title`, `link`, `description`. La syndication désigne le processus qui consiste à publier le même article dans plusieurs journaux, magazine ou sites web. Plusieurs versions ont suivi la RSS 0.91 et l'ont enrichie. Ce fichier standard est mis sur une adresse Internet et permet à d'autres sites d'en récupérer le contenu et de l'afficher sous forme de nouvelles sur lesquelles il est possible de cliquer pour découvrir des documents plus consistants.

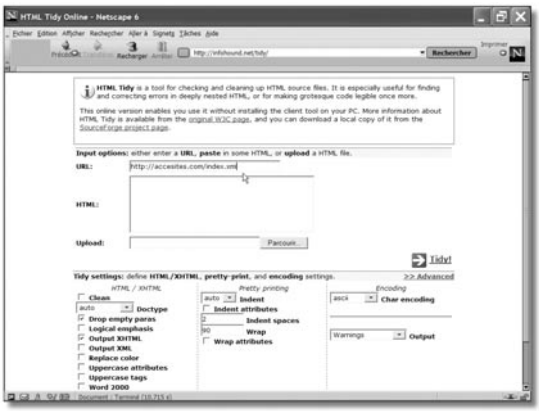
`http://www.pointblog.com/index.rdf`

9.8 Conclusion

Le XML est en train de devenir le vecteur universel de communication et de structuration des données. PHP 5 se tourne résolument vers le XML et propose par ailleurs l'application *Tidy*, qui traite, diagnostique et nettoie le HTML pour le soumettre aux normes du *XHTML*.

▲ Figure 9.3 : *Tidy*

Cette application existe aussi en ligne pour corriger vos fichiers.

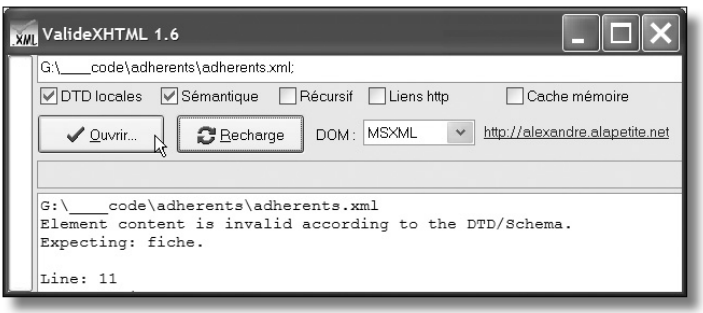


▲ Figure 9.4 : *Tidy en ligne*

Une application gratuite *Valide-XHTML*,créée par Alexandre Alapetite, valide vos fichiers *XHTML*, *XML*, *XSLT* et *DTD*.

▲ Figure 9.5 : *Valide-XHTML*

Valide-XHTML vous précise les erreurs avant de mettre vos fichiers en ligne.



▲ Figure 9.6 : *Valide-XHTML en action*

La plupart des langages ont, implémentés dans leur offre, un ou plusieurs modules XML (*Java*, *PHP*, *Python*, *C#*, *C*, *ASP*...).

◄ Figure 9.7 : *Python*

Cette galaxie XML (*XML*, *XSLT*, *XPath*, *XMLFO*...) est désormais devenue incontournable pour permettre à la grande hétérogénéité de l'informatique de converger.

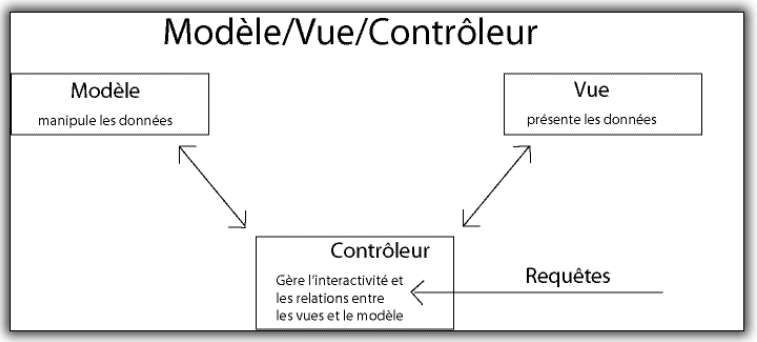
10

Les frameworks ou cadriciels PHP

Le but de ce chapitre n'est pas de donner un mode d'emploi exhaustif, ni un panorama complet des plateformes PHP. Il vous permettra juste de gagner du temps après une veille technique assez poussée pour trouver des plateformes compatibles PHP4 et PHP5, qui respectent le protocole Modèle/Vue/Contrôleur et sont suffisamment développées pour répondre à un grand nombre de situations.

Les autres critères importants dans ce choix sont la documentation et l'aide. L'aide est donnée par des forums avec un grand nombre d'utilisateurs susceptibles de vous répondre. Parfois, comme avec Xataface, l'auteur lui-même se rend disponible pour répondre, voire modifier le code ou ajouter des fonctionnalités qui lui sont suggérées. Ce suivi est un gage de durée et d'évolution.

La description de chaque application sert donc ici à donner une idée claire de ce dans quoi vous vous engagez en adoptant telle ou telle plateforme.



▲ Figure 10.1 : *MVC*

Pour nos besoins courants, nous utilisons trois plateformes écrites en PHP. Ce choix est le résultat d'une veille technologique et de différents tests sur des plateformes diverses :

- *Xataface* pour nos applications simples en interne ;
- *CakePHP* pour nos applications complexes ;
- *Joomla* ou *Drupal* pour les sites Internet ouverts au public.

Une plateforme (ou cadriciel ou framework en anglais) est une sorte de coquille vide dans laquelle il est possible de créer son application. Ces trois plateformes sont compatibles avec PHP4 et PHP5.

10.1 Une méthodologie de travail

Cette structure impose une méthodologie de travail, un cadre de développement, par son arborescence de fichiers et de répertoires. Ainsi la conception de ces plateformes est développée selon le schéma Modèle/Vue/Contrôleur (MVC), les données (Modèle) étant séparées de la présentation (Vues) et des applications qui contrôlent ces données (Contrôleur). Chacun des fichiers entre alors dans une de ces fonctions et se range ainsi dans un répertoire donné. Ce cadre facilite la collaboration entre plusieurs développeurs ainsi que la maintenance de l'application qui obéit à des normes. Le nommage des fichiers et leur place obéissent à des conventions. Ainsi l'application connaît implicitement la fonctionnalité des fichiers selon sa place et son nom, sans qu'il soit obligé de le préciser dans un fichier de configuration volumineux. Une fois ces conventions assimilées, la création d'un site est très rapide.

L'inconvénient de la plateforme est qu'il faut quasiment apprendre un nouveau langage ; la documentation y est donc stratégique.

La plateforme propose toutes les fonctions de base communes à toute application comme la création de menus de navigation ou les formulaires. Ces routines sont souvent fastidieuses à écrire et le fait de ne pas avoir à les recréer ex nihilo permet de gagner du temps.

Plus l'application est standard, plus cela devient simple car il suffit de se conformer à la documentation de base. Plus il faut ajouter de fonctionnalités, plus la documentation est importante. Cette documentation est fournie par le site qui propose la plateforme, par un ou plusieurs forums et parfois, par l'auteur ou les auteurs. Plus la plateforme est utilisée, plus la documentation est fournie et plus le forum est prolixe, ce qui simplifie l'élaboration d'une application. Le forum est là pour résoudre les problèmes des utilisateurs, comme une hotline ; il s'enrichit donc au fur et à mesure de nombreuses études de cas.

Codage des caractères

Ces plateformes proposent différents codages de caractères entre l'UTF-8 et l'ISO-8859-1. L'UTF-8 permet d'une part l'internationalisation du site et autorise d'autre part l'utilisation de caractères français comme le œ. C'est donc préférable. Les données doivent être codées en UTF-8 ainsi que la page en PHP. La plupart des logiciels d'édition de code permettent le codage des caractères en UTF-8 mais provoquent des erreurs PHP. PSPad (`www.pspad.com/fr`), un logiciel gratuit, permet de coder une page sans ces erreurs.

Cette structure est conçue également pour accueillir des thèmes graphiques ou patrons (templates) qui donnent un aspect original à un site ou à une application. Les fichiers de template sont souvent faits sur le modèle de Smarty et comme Xataface qui utilise par défaut les graphismes open source de Plone (`http://plone.org/countries/fr`). Les patrons sont des fichiers HTML avec des parties code accompagnées de fonctions propres à la plateforme.

10.2 L'architecture Modèle/Vue/Contrôleur

L'architecture MVC est constituée de trois composants qui jouent des rôles complémentaires.

Modèle

Le modèle est tout ce qui accède aux données brutes de la base de données. Le modèle encapsule et manipule les données depuis et vers la base de données, les extrayant, les ajoutant, les modifiant et les supprimant. Toutes ces méthodes sont les routines pour agir sur les données. Ainsi le modèle sera le seul code à modifier en cas de déménagement du site sur un nouveau serveur.

Vue

La vue est la part du composant qui sert à présenter les données depuis le modèle, de manière intelligible pour que l'usager puisse interagir. C'est

généralement la page HTML ou le template (patron). La vue reçoit les données du modèle pour les placer à bon escient dans le patron afin de créer des formulaires avec des menus de sélection, créer des articles... La vue ne modifie pas les données mais elle se contente de les mettre en forme en créant une page.

Contrôleur

Le contrôleur répond en cas d'interaction de la part de l'usager. Ainsi dans une application, en général, c'est toujours la même page qui est appelée mais c'est le contrôleur qui décode les données dans l'URL – comme `http://www.monsite.com/index.php?page=2,` – et qui les envoie au modèle, pour qu'il y réponde, puis récupère la réponse pour l'envoyer à la vue qui la met en forme.

10.3 Xataface, donnez un visage humain à votre base de données

Xataface, anciennement nommée Dataface, est une couche applicative qui recouvre une base de données MySQL. La mise en place est simple. D'autres applications existent comme PhpMyAdmin mais avec Xataface, l'interface est plus ergonomique et l'application plus versatile, permettant d'ajouter des fonctions et des interfaces graphiques pour que tout un chacun puisse l'utiliser et constituer un vrai site web. Un système d'authentification assure la confidentialité des données et la personnalisation de l'accès aux différentes tables et champs de formulaire générés.

◄ Figure 10.2 : *Xataface*

Avec Xataface, vous n'avez pas besoin de créer les menus, les listes et les formulaires. Il est juste nécessaire de configurer quelques fichiers texte et votre site est créé. Ensuite libre à vous de personnaliser le résultat.

L'application se trouve sur `http://xataface.com` et s'installe dans un répertoire du serveur. Ce seul dossier peut avoir plusieurs applications en satellite dans des répertoires voisins hors du répertoire maître.

CRUD

CRUD signifie *Create*, *Read* (ou *Retrieve*), *Update* et *Delete* (ou *Destroy*) : créer, lire, mettre à jour et supprimer. Le principe du CRUD est de créer de nouvelles données par un formulaire, de les afficher sous forme de listes ou de fiches, de les modifier par le même formulaire et de les supprimer. Tout cela dans la même interface.

Xataface obéit au principe du CRUD. La programmation est une programmation objet, la mieux adaptée pour une application de ce type.

Xataface est très rapide et simple à mettre en œuvre. D'ailleurs l'auteur, Steve Hannah, propose un tutoriel vidéo sur le site qui explique le tout en quelques minutes, en anglais pour l'instant mais le visuel est très parlant.

Pour commencer, il faut installer le répertoire Xataface sur le serveur et y insérer la dernière version. Il ne faut pas toucher à ce répertoire ; c'est une sorte de boîte noire car votre application est dans un autre répertoire, quel qu'il soit.

Pour modifier les fichiers de son application, il n'est pas nécessaire d'aller changer les fichiers de Xataface, c'est même déconseillé. Il suffit de créer d'autres fichiers avec le même nom et la même arborescence dans le répertoire de sa propre application. En quelque sorte, les fichiers dans l'application surdéfinissent les fichiers de la plateforme elle-même.

◄ Figure 10.3 : *Xataface*

L'arborescence comprend les répertoires *tables*, *conf* et *actions* et quelques fichiers : *index.php*, *conf.ini* et *actions.ini*.

L'architecture

Par l'architecture des fichiers, il est possible de surdéfinir un fichier de Xataface, en en créant une copie dans l'application. Ainsi le cœur de Xataface contient un dossier *templates* qui contient les patrons donnant

leur apparence aux pages ou à des parties de page. Il suffit de créer un dossier de même nom dans l'application particulière en y insérant un fichier template modifié, avec le même nom de fichier, pour qu'il soit utilisé en lieu et place de l'original.

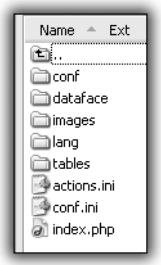
Index.php

Le fichier *index.php* indique la vue par défaut de l'application, le chemin vers le fichier *xataface-public-api.php* et l'URL vers le répertoire de Xataface sur le serveur. Ensuite, il charge une instance de l'objet Xataface et l'affiche dans l'interface standard ou une autre interface :

```
$time = microtime(true);
if (!@$_GET['-sort'] and @$_GET['-table'] == 'fichestravaux'){
$_GET['-sort'] = 'moment desc';
$_REQUEST['-sort'] = 'moment desc';
$_GET['-table'] = 'fichestravaux';
$_REQUEST['-table'] = 'fichestravaux';
}
if (empty($_GET['-action'])) $_GET['-action'] = "list";

require_once '/var/www/html/dataface-0.7.1/dataface-public-api.php';
// include the initialization file
df_init(__FILE__, 'http://lintranet/dataface-0.7.1');
// initialize the site

$app =& Xataface_Application::getInstance();
// get an application instance and perform initialization
$app->display();
// display the application;
?>
```

conf.ini

Le fichier *conf.ini* est un fichier texte qui contient la configuration générale comprenant le codage des caractères, la langue, les coordonnées de connexion à la base de données MySQL, les tables qui s'afficheront, les coordonnées de l'authentification et les préférences. Chaque partie dans ce fichier est introduite par le nom entre crochets : `[languages]`, `[_database]`, `[_tables]`, `[_auth]`, `[_prefs]`. Ces 5 parties gèrent respectivement la langue du site, les données de connexion de la base de données, les tables qui seront affichées, la table et les champs utilisés pour l'authentification et les préférences d'affichage.



◄ Figure 10.4 :
Exemple d'arborescence

Les tables

Ensuite, la partie la plus importante est celle des tables. Le répertoire *tables* contient un répertoire pour chaque table, chaque répertoire portant le nom de la table.Ces répertoires contiennent 3 fichiers texte avec une extension *.ini*.

fields.ini

Le premier, *fields.ini*, est attaché aux champs du formulaire qui attaqueront la table pour agir sur les données. Ce fichier permet de choisir les champs qui seront affichés, d'agir sur leur nature, sur leur mode d'affichage, sur l'ordre d'affichage...

Le fichier *.ini* est un fichier texte où les lignes commençant par un **;** sont des commentaires.

relationships.ini

Le deuxième, *relationships.ini*, contient les modalités de relation entre la table et les autres tables (clé étrangère) :

Le code est de type :

```
table.id = "$id_table"
```

Ici le premier membre désigne la clé étrangère de la table en relation et le deuxième membre concerne la clé de la table courante.

Ou alors, il est possible d'y inscrire une requête SQL qui identifie la relation entre les deux tables.

```
__sql__ = "SELECT * FROM table1, table2 WHERE table1.id= table2.id_table1"
```

Plusieurs relations peuvent ainsi être définies sur ce même fichier, chaque table étrangère étant identifiée par [table1], [table2]...

valuelists.ini

Ce fichier regroupe les valeurs des champs de sélection (select, checkbox, boutons radio...) et celles des champs autocomplétés.

Autocomplétion

L'autocomplétion est cette fonction qui, quand on commence à saisir un mot dans le champ d'un formulaire, fait apparaître une liste de mots commençant par les mêmes lettres et qui ainsi peuvent être validés comme valeur du champ.

Ces valeurs sont données sous forme de liste ou par une requête SQL sur une valeur :

```
[select_OUI]
__sql__ = "SELECT clef, libelle from select_OUI"
[Creneaux]
M = Matin
A = Après-midi
J = Journée
```

Classe pour la table

La classe de la table peut être créée quand le besoin de fonctions originales se fait sentir. Ainsi il peut être nécessaire de formater les dates dans les champs *date* ou d'afficher une valeur par défaut dans un champ quelconque.

Le fichier prend le nom de la table *table.php*, *table* étant mis pour le nom de la table et donc du répertoire où le fichier se trouve.

```
class tables_repertoires {
    function after_action_new(){
            header('Location: '.'http://www.monsite.net/admin/
           index.php?-action=list&-table='.@$_REQUEST['-table']);
            exit;
    }
}
```

Les fonctions ont des conventions de nommage dans la classe. Ainsi, par exemple, cette fonction est déclenchée une fois qu'on a créé une nouvelle entrée dans une table quelconque comme *repertoires*. Cette fonction, dans cet exemple, affiche la liste des répertoires dans la table. De nombreuses fonctions existent, en particulier pour afficher une date avec un format spécifique, ou afficher par défaut les valeurs d'un champ, avec, à chaque fois, une convention de nommage dans le nom de la fonction.

Application Déléguée dans le répertoire conf

L'application **Déléguée** est une application qui agit sur toutes les actions déployées. Ainsi ce fichier peut contenir l'authentification. Il se trouve dans le répertoire *conf* :

```
class conf_ApplicationDelegate {
     function getPermissions(&$record){
         $auth =& Dataface_AuthenticationTool::getInstance();
         $user =& $auth->getLoggedInUser();
         if ( !isset($user) ) return
        Dataface_PermissionsTool::NO_ACCESS();
         $role = $user->val('role');
         return Dataface_PermissionsTool::getRolePermissions($role);
      }
}
```

Cette fonction, ici, identifie chaque utilisateur et retourne la permission qu'il détient. Il est possible d'affiner la granularité en définissant les accès aux tables et même aux champs d'un formulaire qui dès lors que l'usager n'y aura pas accès, ne sera pas actif.

Conclusion

Xataface permet de monter un site en quelques minutes. Ce qui prend le plus de temps est de changer l'interface graphique mais si vous créer juste une application pour quelques personnes, elle sera opérationnelle immé-

diatement. Une autre utilisation possible consiste à créer toute l'administration d'un site qui lui, se sert de la base de données gérée par Xataface. C'est une plateforme dont on ne peut plus se passer une fois qu'on y a goûté. La documentation est encore en anglais mais l'application a été traduite et la prochaine version sera plus aboutie, avec une traduction plus complète et une documentation en français.

10.4 CakePHP, c'est du gâteau

Le projet CakePHP (`www.cakephp.org`, `www.cakepourtous.org`) a démarré en 2005, avec comme ambition de proposer un Ruby on Rails pour PHP.

Ruby on Rails ou RoR

Ruby on Rails est sorti en juillet 2004 et a été créé à partir du langage Ruby qui est un langage de script orienté objet. La sortie de RoR fut un coup de tonnerre dans la communauté des développeurs car il répondait exactement aux exigences d'interactivité du Web 2.0. Il permettait de créer rapidement un site avec du code concis, peu de configuration et de l'ajax pour la convivialité. Il est utilisé par un certain nombre de sites web 2.0.

L'arborescence reflète parfaitement ce modèle. Le répertoire *cake* est le répertoire auquel le développeur ne touche pas, excepté quand il fait une mise à jour pour passer à la version supérieure de Cake. L'autre répertoire important est *app* qui contient l'application avec les dossiers *models*, *controller* et *views*.

◄ Figure 10.5 : *Cake*

Chaque partie de l'application doit avoir sa part de modèle, de contrôleur et de vue. À ces répertoires s'ajoutent le répertoire *tmp* qui contient les

fichiers de cache, de sessions, de logs et tous les fichiers temporaires. Le dossier *config* contient les rares fichiers de configuration dont les réglages pour la base de données. Le dossier *vendors* sert à ajouter des scripts à l'application. Un autre dossier *vendors*, situé en dehors du répertoire *app*, rassemble les scripts génériques destinés à toutes les applications.

▲ Figure 10.6 : *www.cakephp.org*

Cake offre les mêmes fonctionnalités que Xataface, avec un plus large spectre (différents types de bases de données) mais aussi plus de complexité. Cake simplifie l'usage des bases de données et il présente un certain nombre de caractéristiques.

Cake utilise le *mod_rewrite* d'Apache pour réécrire les URL à la volée. Ainsi l'URL `http://localhost/club/edit/1/` ira chercher le contrôleur `club`, la méthode `edit()`, le modèle pour aller trouver la donnée avec l'id 1 puis l'affichera dans la vue *edit.thtml*. C'est très logique et pour utiliser cette application, il suffit de se conformer à ces conventions.

En ajoutant la variable `$scaffold` (échafaudage) dans le contrôleur, Cake crée les fichiers par défaut. Il suffit ensuite de les modifier pour les rendre conformes au résultat désiré.

```
var $scaffold;
```

Sous le dossier *app*, il reste deux dossiers importants : *webroot* qui contient tous les fichiers qui apparaissent en vitrine (images, css, javascript...) et le dossier *helpers*. Ces helpers ou aides sont des fichiers PHP qui modifient l'affichage d'un élément dans les vues. Il existe des helpers natifs pour Cake comme les helpers HTML qui permettent au développeur d'écrire son code très rapidement. Ainsi pour écrire le code d'une aire de texte :

```
$html->textarea('Note/body', array('cols'=>'60', 'rows'=>'10'));
```

La notion de rappel (*callback*) qui a déjà été vue pour Xataface intervient. Ainsi une action déclenche une nouvelle action qui se déroule avant ou après.

Cake est une application copiée sur Ruby on Rails ; la conception est dérivée de Ruby contrairement à Xataface qui est plus axée MySQL et PHP et respecte davantage l'esprit de Ruby on Rail dans un sens de légèreté.

10.5 JOOMLA, le meilleur gestionnaire de contenu

Joomla (`www.joomla.fr`) est le meilleur gestionnaire de contenu car il est très complet et utilisé largement. Ainsi, il enrôle de nombreux développeurs et contient de nombreuses fonctionnalités pour répondre à diverses situations. Joomla est né d'une scission avec la plateforme Mambo. Joomla signifie "*tous ensemble*" en Swahili.

◄ Figure 10.7 : *Joomla*

Le principe de ce type d'application est que le contenu est géré séparément de la présentation :

- Ils permettent à plusieurs rédacteurs de travailler sur un même document.

- Ils suivent un document de la conception à la publication.
- Ils regroupent et gèrent des documents de nature différente (articles, documents à télécharger, formulaires, FAQ, calendriers d'événements...).

Joomla permet de séparer la technique web de la rédaction de billets. Ainsi un site Joomla a un webmestre qui s'occupe de la technique et plusieurs rédacteurs pour rédiger le contenu. Les rédacteurs sont aidés par des traitements de texte avec leur barre de texte et des mini-calendriers pour choisir une date et l'insérer dans un bon format.

▲ Figure 10.8 : *Joomla*

À une plateforme commune s'agglutinent des extensions, petites applications qui ajoutent des fonctionnalités au tronc commun.

Joomla est donc un standard avec une grande diversité en raison de ses extensions. Cette diversité explique aussi la possibilité de failles de sécurité ou d'impasses, l'extension n'étant plus mise à jour au gré des nouvelles versions de Joomla.

Permissions

Joomla propose un diagnostic de votre site à l'installation. Un certain nombre de répertoires demandent une permission 0777 pour l'installation. Veillez à remettre 0755, une fois l'installation finie.

Les extensions personnalisent Joomla. Elles sont de quatre types :

- *Les composants* sont des mini-applications, agissant aussi bien en avant-scène (*frontend*) qu'en coulisses (*backend* ou administration) et rattachées souvent à une ou plusieurs tables de la base de données.
- *Les modules* sont des éléments d'interface qui se trouvent généralement dans les colonnes de droite et de gauche dans la page d'accueil. Ils présentent souvent une information, des liens d'une manière différente pour les mettre en valeur.
- *Les mambots* sont des fonctions qui sont actionnées par des événements et permettent de transformer les informations avant leur affichage.
- *Les templates* (ou patron, comme des patrons de couture) sont la peau dans laquelle se glisse l'application. Un patron est composé d'un fichier PHP, d'images et d'un ou plusieurs fichiers CSS pour régler l'apparence de votre site.

Choisir Joomla 1.0.15 ou 1.5

La version 1.0.15 est la plus ancienne et propose 2 700 extensions qui fonctionnent. La version 1.5 est internationalisée et elle est normalisée pour un développement. Si les extensions sont primordiales, il vaut mieux adopter l'ancienne version. Si c'est pour un nouveau site où les extensions ne sont pas fondamentales, il vaut mieux passer au 1.5.1 directement. Le code a été simplifié et normalisé. Son développement a demandé deux ans de travail.

Drupal contre Joomla

◄ Figure 10.9 : *Drupal*

Un autre CMS très complet et bien écrit est Drupal (`http://drupalfr.org/`).Son code est mieux écrit et répond mieux aux normes de codage. Il a cependant moins de thèmes et d'extensions disponibles.

S'il fallait départager les deux, il faudrait tenir compte de l'utilisation et du type de site.

Drupal 6.8 permet :

- de créer un site multi-sites et donne plus de latitude pour moduler les catégories ;
- de déployer une granularité plus fine des permissions ;
- de modifier facilement les quelques thèmes pour les rendre originaux ;
- de monter un site communautaire avec de nombreux membres ;
- de l'intégrer dans d'autres sites grâce à son code aux normes.

Joomla 1.5.8 est :

- facile à comprendre, à installer et à enrichir ;
- implémenté avec un espace d'administration ergonomique ;
- parasité par des interactions entre les modules (SEF…) et des bogues dû à un code mille-feuilles ;
- difficile à intégrer à d'autres sites ;
- riche d'innombrables modules et extensions.

10.6 Conclusion

▲ Figure 10.10 : *Ruby on Rails*

Les plateformes sont des aides précieuses pour créer de nouveaux sites, collaborer et maintenir un site web évolutif mais elles ne peuvent servir pour reprendre un site déjà fait. PHP est un langage de script très souple et un site web peut être créé de multiples façons dans l'architecture et la syntaxe ; les cadriciels imposent un certaine "normalité" dans le code et la méthode, ce qui rend plus facile la maintenance et accélère les temps de développement, une fois les conventions assimilées.

Certains d'entre vous souhaiteront aller boire à la source et utiliseront Ruby on Rails (`www.rubyonrails.org`) et le langage Ruby (`www.ruby-lang.org/fr/`), d'autres employant le PHP5 s'orienteront vers Symfony (`www.symfony-project.org`) mais le système reste le même et le langage PHP a tout à gagner dans le développement et le foisonnement de ces plateformes.

11

Mettre de l'ordre

Vous voilà maintenant avec un certain nombre de fichiers et d'applications codées sur des centaines de lignes. Vous pouvez continuer à créer de nouvelles applications, si vous aimez concevoir et développer, ou choisir de vous pencher sur l'architecture de vos applications et leur maintenance, pour construire vraiment et améliorer les fichiers au fur et à mesure que vous progressez. Vous changerez ainsi le code circonstanciel en code élégant, réutilisable et explicite.

L'application idéale pourra être placée sur n'importe quel serveur et fonctionnera sans modification. Si des changements sont nécessaires, créez des variables que vous grouperez en tête de fichier. Une autre solution, si vous avez de nombreuses variables à modifier, consiste à créer un fichier de configuration que vous incluez en tête d'application.

Personne n'aime se replonger dans des vieilleries pour les trier. Aujourd'hui, votre application est nouvelle et vous codez avec lyrisme, emporté par l'enthousiasme. Tout vous paraît clair. Telle variable possède une utilité évidente, telle autre s'articule avec telle donnée. Mais demain, un de vos utilisateurs trouvera peut-être un bogue, vous aurez appris de nouvelles façons de coder du langage PHP ou la nouvelle version du moteur PHP sera sortie et vous aurez envie d'uniformiser votre code, ce qui n'est pas forcément une bonne idée. Pourquoi toucher en effet à une application qui remplit son rôle ?

Ce jour-là, vous comprendrez à peine la logique qui a présidé à vos applications construites sans méthode. Pour qu'une application vive, elle doit être conçue avec soin.

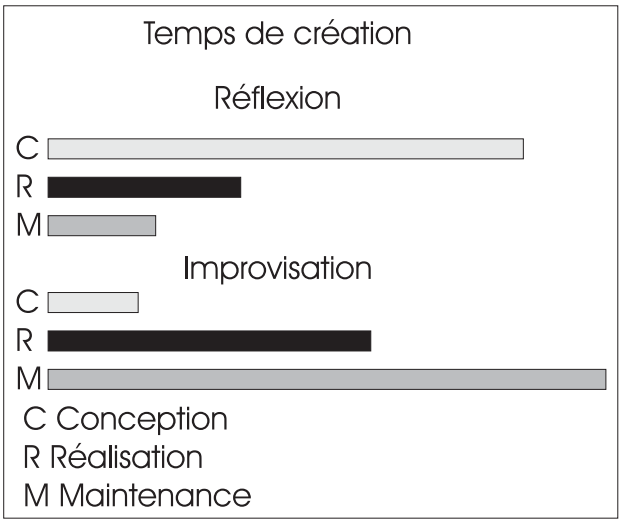
Utilisez des conventions qui ne varient pas et vous donnent des repères. Par exemple, les noms de variables `$resultat`, `$passage` ou `$rang`, que vous employez dans chaque application, créeront des repères dans des applications de milliers de lignes. Aménagez des ruptures dans vos pages avec des barres obliques de commentaires pour marquer les limites d'un module indépendant. Commentez abondamment pour que la logique sous-jacente apparaisse. Expliquez vos choix. Donnez des noms explicites à vos variables et à vos fonctions. Au besoin, créez de la documentation technique. Sa rédaction vous obligera à préciser des points flous. Elle sera un élément déterminant pour communiquer avec d'autres développeurs.

À certains endroits de votre application, vous aurez des idées de développement, de fonctionnalités supplémentaires pour l'utilisateur mais il se peut que vous soyez pressé par le temps. Si vous ne pouvez réaliser votre idée, prenez soin tout de même de l'exposer clairement dans vos commentaires ou une documentation à part.

Créez un document qui répertorie les fonctions que vous avez conçues avec leur description. Imprimez-le et placez-le à votre portée.

L'élaboration d'une application se fait en trois étapes : la conception et l'architecture, la réalisation, les tests. Ces trois étapes sont réalisées par une personne, ou trois ou quatre personnes disposant d'une plus grande spécialisation. Les termes varient selon les entreprises et l'aspect privilégié pour chaque tâche.

11.1 Avant



▲ Figure 11.1 : *Importance du travail préparatoire*

Essayez de prendre quelques jours de réflexion avant d'entrer dans votre application. Imaginez les différents moments de votre application, esquissez quelques lignes de code sans les développer. Explorer les directions

possibles à chaque aiguillage de votre arborescence. Si vous travaillez pour quelqu'un, essayez de lui faire préciser ses objectifs et les contraintes incontournables. Par exemple pour le magazine, il est intéressant de savoir combien de pages peut faire un article ou une rubrique, combien d'images, comment elles sont placées, comment est faite l'accroche… Préparez vos questions. La clarté d'une réponse dépend de la précision de la question. La relation au client est l'art de poser les questions avec des termes précis, clairement compris par l'interlocuteur.

11.2 Après

Après la conception de l'application, des tests en profondeur sont indispensables. D'abord pour déceler les bogues qui peuvent se nicher dans des recoins insoupçonnés, aussi pour emprunter toutes les avenues ouvertes à l'internaute et vérifier que pour chaque action, une réponse est prévue. Votre application n'est pas linéaire, elle est arborescente et vous devez, en tant qu'auteur, visiter la moindre ramure. Dessinez un arbre de possibilités et certaines éventualités oubliées vous sauteront aux yeux. Rendez les pages d'erreur agréables à l'œil. Reprenez les fonctions qui retournent un résultat ou un entier booléen et créez un bloc d'exécution pour chaque erreur d'exécution. Incluez une fonction `mail()` qui vous envoie un courriel avec la variable `$PHP_SELF` pour avoir l'identité du fichier et le type d'erreur rencontré.

N'oubliez pas de tester votre communication auprès de "candides". Soyez explicites dans vos indications sans noyer l'internaute avec des directives trop longues. Ceci fait l'objet du chapitre 11.

11.3 Le débogage

Faire une erreur est facile, la déceler peut parfois demander beaucoup de temps et d'effort. Il existe plusieurs types d'erreur pour un programmeur. Cela va de la simple faute de frappe à une inversion de paramètres dans une fonction, en passant par une confusion de nom de variable.

Les plus faciles à déceler donnent le numéro de la ligne fautive en précisant la cause du problème. Parfois, cette indication est une fausse amie car elle indique la dernière ligne de votre application, ce qui constitue souvent une erreur d'accolade, de parenthèse ou de guillemets.

Une astuce pour les corriger consiste à faire une recherche multiple avec votre éditeur de code sur chaque élément de la paire pour les accolades et les parenthèses. Si le nombre de parenthèses ouvrantes et celui des parenthèses fermantes ne sont pas identiques, le problème se situe à cet endroit. Pour les guillemets, en général, la ligne faussement accusée est un peu plus bas que la fautive. Remontez et vous trouverez un guillemet manquant, un guillemet double qui répond à un guillemet simple, un guillemet interne sans signe d'échappement ou un point-virgule manquant. Parfois ce ne sera pas si facile.

Vos fonctions ne doivent pas contenir l'opérateur `@` qui empêche l'affichage de l'erreur. Vous l'ajouterez une fois les tests effectués, en prenant soin de mettre un dispositif qui vous prévienne de l'erreur avec une identification claire du fichier et de l'erreur.

Le débogage est d'abord une analyse. Elle consiste à séparer les éléments qui composent l'application afin de déceler la cause de l'erreur : séparer les éléments dans le temps et dans l'espace du code.

Si vous avez déjà testé le code et qu'il fonctionnait jusqu'à tel moment, remontez la chronologie. La confusion vient souvent du fait que nous avons accompli plusieurs actions en même temps et que nous pouvons partir dans une mauvaise direction et compliquer la recherche. L'erreur s'est-elle produite quand vous avez changé de machine ou de serveur, ou simplement changé la configuration, ou la version de PHP ?

Vérifiez l'ordre des paramètres dans les fonctions appliquées sur les chaînes de caractères ou sur les expressions rationnelles.

Tab. 11.1 : Ordre des paramètres

Fonction	Place des éléments
`explode()`	Séparateur, texte
`split()`	Motif, texte
`ereg()`	Motif, texte
`ereg_replace()`	Motif, texte de provenance, texte de destination
`strchr()`	Texte, motif
`strstr()`	Texte, motif

Lorsque vous êtes certain que le problème vient du code, analysez votre code et clarifiez-en la structure en créant des blocs séparés par des lignes de barres obliques. Ensuite, appuyez-vous sur le code essentiel en désactivant les lignes qui ne font qu'améliorer votre algorithme. Pour les désactiver, mettez-les en commentaires, c'est-à-dire faites-les précéder de deux barres obliques. Le moteur PHP les prendra pour des lignes de commentaires et ne les exécutera pas. Dans le code restant, affichez la valeur des différentes variables, du début vers la fin. Nous avons suggéré précédemment de créer des ruptures dans vos applications pour délimiter les modules. Créez des points d'arrêt pour connaître la teneur des variables en sortie de traitement avec des

```
echo $variable;
exit;
```

ou à l'intérieur des boucles avec `exit` ou `break`

```
echo "$variable<BR>";
break;
}
exit;
```

Ajoutez des `return` dans vos fonctions, ils ont pour effet d'arrêter l'exécution de la fonction et de retourner une valeur.

Une fois que tout fonctionne comme vous le désirez, réactivez votre code secondaire par étapes. L'erreur peut provenir d'une partie du code qui neutralise l'autre partie. Si vous êtes dans les fonctions de systèmes de fichiers, vous voulez par exemple obtenir le `filemtime` en pensant obtenir le `TIMESTAMP` de la dernière modification du fichier. En fait, juste avant, vous devez utiliser le `chmod()` pour changer les propriétés du fichier. Cette commande va aussi modifier le `filemtime()`. Votre logique serait contradictoire.

Vérifiez vos blocs conditionnels. Avez-vous un signe `=` ou `==` ? Les signes > ou < peuvent être à l'envers.

Soyez méticuleux sur les noms de variables et les chemins de fichier. Une lettre manquante ou supplémentaire, et votre variable ne renverra aucune valeur, votre programme ne trouvera pas le fichier ciblé. Assurez-vous que le fichier ou le répertoire possède les permissions requises (`chmod`).

Si vous n'avez rien sur l'écran, affichez le code source dans le navigateur. Soit vous avez un code HTML tellement corrompu, notamment avec des tables, que le navigateur n'affiche rien, soit votre requête SQL contient une erreur et ne retourne aucun résultat. Souvent, le moteur PHP écrit une phrase d'erreur dans le code source qui n'apparaît pas dans la fenêtre du navigateur. L'erreur peut générer également un temps d'exécution trop important.

Si vous avez une erreur d'affichage HTML, vous pouvez récupérer le code HTML obtenu et l'imprimer pour en voir les erreurs ou le mettre dans un éditeur HTML `wysiwyg`, comme Dreamweaver (sigle de `What You See Is What You Get` : ce que vous voyez est ce que vous obtenez).

Pour une erreur de requête SQL, lancez la requête dans PhpMyAdmin pour déceler la cause. Si une requête n'est pas comprise par la base, vérifiez le nom des colonnes et la place des clauses. Simplifiez votre requête et testez-la en ajoutant les clauses peu à peu. Les clauses `LIMIT` ou `ORDER BY` sont toujours après la clause `WHERE`. Pour une insertion, avez-vous le même nombre de valeurs que de colonnes citées ?

11.4 Sachez anticiper

Nous avons vu les fonctions, les variables communes, les constantes, l'authentification, la connexion à la base de données et les statistiques. Vous pouvez ajouter une fenêtre de publicité ou une gestion de vos bannières, une personnalisation de toutes les pages du site, y compris les pages qui ne font pas partie du club, pour tous ceux qui utilisent le cookie… Ces idées sont intéressantes. Elles vous demanderont de créer d'autres applications pour vos pages. Si vous avez de telles idées avant de créer votre site, vous gagnerez du temps et serez plus efficace. Avec la méthode des fichiers inclus, l'ajout d'applications supplémentaires sera cependant facilité. Faites en sorte d'avoir le moins de données à ajouter pour chaque application, comme dans le mini-forum où le nom de la table prend le nom du fichier en enlevant le chemin et l'extension au moyen d'une expression rationnelle. Utilisez au maximum des fonctions PHP pour avoir peu de paramètres à approvisionner. Dans vos applications, laissez la porte ouverte à des extensions de programmes afin que l'ajout d'applications ne commence pas par des tâches fastidieuses.

12

La traversée de l'écran

l'Internet n'est qu'un outil au service de la communication et de la créativité. Nous communiquons avec un grand nombre de personnes à la fois, de tous les pays, mais dans le même temps, la communication passe mal d'un individu à l'autre car cet outil informatique n'est qu'une prothèse pour pallier notre manque de communication. Quand nous entrons sur un site qui nous interpelle par notre nom, nous sommes dans le virtuel. Nous croyons à une reconnaissance mais personne ne doit être dupe de ce jeu. Nous ne sommes pas dans la réalité. Vous ne réussirez la communication sur votre site qu'en traversant l'écran. Pour cela, il faudra vous impliquer personnellement. Vous ne pourrez avoir une relation personnelle avec tous les membres de votre site, là n'est pas le but, la création d'un noyau de fidèles suffira. Vous n'avez pas besoin d'entretenir une relation suivie mais si vous communiquez par courriel de loin en loin, par causerie ou visiophonie, ou même si vous organisez des rencontres dans des lieux réels, vous créerez une véritable relation, quel qu'en soit le degré d'intimité.

Le partage constitue parfois l'un des buts du site Internet mais beaucoup de sites demeurent malheureusement un prolongement de l'ego, alors que toute communication réelle est un échange qui comprend un temps d'expression et un temps d'écoute. C'est pour cette raison que le langage PHP et les bases de données MySQL ont tant de succès. Ils rendent en effet le site interactif. Ce dernier vous échappe car il appartient à tous ceux qui possèdent les mêmes goûts que vous. De nombreux outils sont présents sur l'Internet et gratuits. Vous voulez un portail personnel et interactif, vous avez l'embarras du choix : *SPIP*, *PHPNuke*, *Thatware*… Le seul problème est que tous ces sites, bien qu'interactifs et vivants, sont formatés sur le même modèle. Il est plus intéressant de maîtriser le langage de construction de ces applications et de les modifier comme on le souhaite.

Au moment de la conception de votre site, réfléchissez à ce qui est commun à vos pages. Quelles sont les données importantes et les fonctionnalités que vous voulez proposer aux internautes ? Comment allez-vous classer les pages ? Un site est un carrefour d'informations. Voyez ces mouvements de données, les communications entre les différents acteurs. Que voulez-vous que la machine fournisse ? Comment faire agir et réagir les internautes ? Comment coordonner des actions de promotion avec des sites complémentaires ? Vous pouvez rédiger une charte qui montre sur quoi vous vous engagez vis-à-vis des membres. Le

site Internet n'est pas une affaire d'informatique mais bien de création et de communication.

Faites appel à votre créativité, laissez-vous aller à rêver votre site, tournez-le dans tous les sens, dessinez-le avec des crayons de couleurs. Voyez-le comme un centre énergétique avec de l'information qui circule. Sortez des sentiers battus. Imaginez votre site comme un volume. Cassez cette fascination pour l'écran et sa vision plate. Tout ceci vous aidera à sortir du conformisme de certaines pages, du jardin à la française et du cordeau cartésien. Une fois votre site dessiné, il sera toujours temps de vous frotter aux outils. Nous connaissons des programmeurs qui n'ont rien à dire. Ils offrent alors de faire des pages Web pour d'autres qui sont allergiques aux programmes et à l'informatique. Ainsi les programmeurs se mesurent à du contenu et la personne créative voit son idée traduite sur le Web. Lorsque cela se fait dans la transparence, tout le monde est gagnant. Le programmeur acquiert un peu de créativité et le créatif un peu du sens de la technologie. Beaucoup d'associations possèdent un site Web qui est le fruit de ce type de collaboration.

Créez un forum. Vous trouverez une liste des nombreux outils disponibles en annexe. Pliez-les à votre goût, améliorez-les même. Définissez des listes, des causeries (chat en anglais), des rendez-vous. Rencontrez dans la réalité vos visiteurs. Interrogez-les au moyen de sondages. Ouvrez une boîte à suggestions. Améliorez constamment la transparence de la navigation sur le site, le classement des pages. Créez des outils qui vous facilitent la mise à jour. Que ce soit un métier ou un divertissement, la passion joue un rôle important.

12.1 La communication sur le Web

Toute communication est fondée sur une action/réaction. Quand l'internaute clique sur un bouton, un lien, une image, il s'attend à créer un événement visible. C'est comme une scène avec des coulisses. La plupart des événements se fomente dans les coulisses, sur le serveur. Il faut créer une réponse pour que l'internaute sache que son action a produit un effet. Le webmestre doit faire croire à l'internaute que ce monde virtuel est réel, ou plutôt lui communiquer sa réalité. Le créateur de site crée des réponses à chaque action de l'internaute. Une fois que l'internaute a cliqué sur le bouton **envoyer**, il ne doit jamais se retrouver devant une page blanche. Le travail du concepteur

de site et du développeur consiste à prévoir les actions de l'internaute et à élaborer une réponse adaptée à chacune de ces actions. Il existe toujours plusieurs possibilités, y compris lors d'actions malfaisantes. L'internaute peut télécharger votre formulaire et modifier vos champs cachés avant d'envoyer des données indésirables, voire destructrices.

Souvent, ce que vous pensez évident ne l'est pas pour l'internaute. Ainsi, vous vous apercevez que tel concours lancé sur votre site n'a aucun succès. En regardant de plus près, vous constatez que la question posée n'est pas suffisamment claire, la récompense insuffisante ou que les questions qui l'accompagnent violent la vie privée de l'internaute. Il existe une ligne rouge que vous ne pouvez dépasser. Vous construisez un rapport de confiance avec l'internaute et cette confiance peut être mise à mal par une intrusion, une maladresse ou une tricherie. Au fur et à mesure que l'internaute se promène sur l'Internet, son discernement s'aiguise et les artifices ne l'amusent plus.

L'humour

La machine ne permet pas de se détendre comme lors d'un contact avec un être vivant. C'est pourquoi il est indispensable d'utiliser l'humour pour casser l'appréhension de l'inconnu. La machine génère du stress. L'Internet permet l'instantané et le rapprochement. Pourtant chacun, face à son écran, est seul. Le webmestre qui réussit à le faire oublier à l'internaute gagnera un fidèle qui participera régulièrement à son site.

Le soutien

En tant que webmestre, vous recevrez des questions par courriel dont la réponse vous paraîtra évidente. Si vous créez une liste de courrier, la question la plus courante sera : "comment résilier mon abonnement ?" Vous savez que la réponse est en bas de chaque message envoyé sur votre liste, mais faites l'effort de répondre gentiment et d'expliquer sans que la personne ne se sente idiote. Combien de fois avons-nous répondu trop impulsivement ou trop vite ? Dans notre culture francophone, le langage du corps, le son d'une voix, le ton, le regard et le geste ont plus d'importance que le contenu de la parole. L'Internet, à travers le courriel, nous enlève ces repères et nous ne sommes pas habitués à communiquer de cette façon. Il faudra du temps pour bien maîtriser ce type de communication.

Nous ne regardons pas un écran comme un texte imprimé sur papier. Sur écran, nous lisons en diagonale car l'écran n'est pas fait pour lire. L'œil se fatigue à lire sur un écran. Il est toujours préférable d'imprimer. Vu la quantité d'informations que nous recevons par l'Internet, il n'est pas possible de tout imprimer, d'autant que les informations sont rapidement obsolètes, chassées par l'actualité ou un nouveau flot de courriels. Soyez patient avec vos visiteurs. Ils vous ont fait l'honneur de s'intéresser à votre site. Ils ne sont pas forcément passionnés par l'informatique. Pour eux, c'est simplement un outil pour communiquer qu'ils ne maîtrisent pas toujours.

Les émoticônes

Certaines icônes transmettent un sentiment :=) . Elles placent un peu d'humanité dans les messages. L'information contenue dans un message n'est pas évidente. Vous savez ce que vous écrivez, vous en connaissez l'intention et le ton. Juste avant d'envoyer votre message, mettez-vous à la place de la personne qui le reçoit, hors contexte. Si le message ne donne pas d'indice sur le ton que vous employez, les mots risquent d'être mal compris. Si le message est trop neutre, le récepteur projettera dans ses mots les sentiments qu'il ressent lui-même et vous les attribuera. Un signe ;=) est toujours sympathique pour montrer que vous n'êtes pas vindicatif. Les courriels entraînent tellement de malentendus. En effet, la communication est immédiate, comme la parole, mais à la différence de l'oral, les mots restent et peuvent être mal interprétés car le récepteur n'a pas d'indications sur votre personnalité.

Ce qui arrive par courriels peut se produire sur un site à un degré moindre. En effet, vous avez des images et des couleurs mais votre contenu peut être gommé par la forme de votre site. La forme de votre site accroît-elle la portée de votre message ? Quelqu'un qui vient sur votre site et ne vous connaît pas percevra immédiatement "ce qui ne colle pas". Soyez très clair sur vos objectifs et n'ayez pas d'arrière-pensées. Il vaut mieux faire deux sites différents que de placer deux messages contradictoires sur un seul site.

La nétiquette

La nétiquette est l'étiquette du Net, l'ensemble des règles qui permettent à une communauté de vivre harmonieusement dans le respect mutuel. Un certain nombre de comportements sont mal perçus. Si vous envoyez des

courriels sans qu'ils aient été sollicités, soyez très courtois. Demandez à la personne de s'inscrire à votre liste de courrier si elle est intéressée et précisez que si ce n'est pas le cas, elle ne recevra plus de courriels de votre part. Faites en sorte de vous identifier clairement. Essayez de savoir si tout le monde peut recevoir des courriels en HTML avec des images. Évitez les fichiers joints de type diaporama, mettez-les plutôt sur votre site en donnant l'adresse pour les personnes intéressées. Si vous respectez les autres, vous serez respecté et construirez une audience pour votre site. Il est facile d'accrocher des passants sur le Web. Ils parcourront votre page d'accueil rapidement puis iront voir d'autres sites.

La meilleure publicité reste le bouche à oreille. Il fonctionne de la façon suivante. Vous mettez une information utile, originale ou séduisante sur votre site. Un certain nombre d'internautes auront à cœur d'informer leurs relations qui, à leur tour, informeront d'autres personnes… La notoriété de votre site dépend de ce bouche à oreille. Il vaut mieux avoir un noyau d'une dizaine d'inconditionnels plutôt que des milliers d'internautes venant voir pendant une semaine une animation flash qui les fait rire. De la même façon, un travail en profondeur avec des informations de qualité confirmera la fiabilité de votre site chez les internautes.

La mise à jour régulière

La création d'un noyau de fidèles autour de votre site est conditionnée par une mise à jour régulière. Il suffit d'aspirer certains sites Web pour les consulter sur son disque dur. Comme ils ne sont jamais mis à jour, il n'y a aucun intérêt à les consulter en ligne. Votre site doit vivre et être nourri par vous et vos membres. Un bon site ne va pas de soi. Communiquez, mettez en avant. Abonnez-vous à d'autres listes de diffusion autour du thème de votre site et diffusez de l'actualité brève en plus de vos informations plus approfondies, afin qu'il y ait toujours quelque chose de nouveau sur votre site.

La navigation

La navigation est essentielle pour que les internautes dépassent votre page d'accueil et s'enfoncent dans les profondeurs de votre site. Un site confus frustre le visiteur. La navigation doit être claire. Entre l'internaute et la page, il ne doit pas y avoir trop de clics. Pour qu'une navigation soit claire, votre thème et vos sous-thèmes doivent être clairs. Si vous avez une

classification claire, le site sera simple à composer et à structurer et vous aurez davantage d'inspiration pour créer son contenu. Plus de visiteurs auront envie d'y participer car ils auront l'impression de connaître votre site comme s'ils l'avaient conçus eux-mêmes.

Le référencement

Une fois la tournée de tous les moteurs de recherche accomplie (*Google*, *Yahoo*, *Nomade*, *Lycos*, *Hotbot*, *Voilà*…), sachez que le meilleur référencement est l'échange de liens. Un des moteurs de recherche favoris actuellement est *Google*. Il place en tête les sites dont on retrouve le lien sur de nombreux autres sites. Pour cette raison, nouez des liens autour de vous.

Utilisez les champs *meta*, surtout *keywords* et *description*. Le titre, les mots-clé et la description de votre site ont la plus grande importance, comme vous l'aurez constaté dans le programme de moteur de recherche inclus dans le chapitre *Moteurs de recherche et expressions rationnelles*.

Enfin, utilisez les référenceurs gratuits. Nous en citons dans l'annexe du livre. En un clic, ces applications envoient l'adresse de votre site et sa description dans les principaux moteurs de recherche ou annuaires de liens. Rien ne sert d'avoir son site indexé dans des centaines de moteurs, il vaut mieux être bien indexé dans ceux que les internautes consultent. Alors créez des liens.

12.2 Conclusion

Après une période de forte spéculation qui a fait grimper puis s'écrouler les valeurs boursières du Web, l'Internet redevient ce qu'il était au départ, un outil de communication. Mais n'oublions pas que la machine reste une prothèse destinée parfois à pallier nos manques. Des rencontres peuvent se faire et se font sur l'Internet, mais sans contact humain, elles n'aboutissent pas à une relation satisfaisante. *PHP* et la base de données *MySQL* constituent des outils simples qui proposent l'interactivité comme réponse à la complexité du monde de la communication.

13

PHP 6 en ligne de mire

A l'heure qu'il est, PHP 6 n'est pas encore achevé mais les annonces déjà faites nous en donnent une idée assez précise. Il représente une étape importante vers une normalisation et une sécurisation du code. La version PHP 4 n'est plus soutenue depuis le 5 février 2008. Le PHP 5, lancé en 2004 et qui est devenu stable en 2008, a permis d'asseoir le langage objet dans l'API. Avec l'avènement de PHP 6, PHP gagne en rapidité et en maturité.

◄ Figure 13.1 : *PHP 6*

La version en développement est téléchargeable sur `http://snaps.php.net/`

L'une des nouveautés qui nous intéressera plus particulièrement est la généralisation de l'unicode. L'unicode permet d'avoir des noms de variable dans toutes les langues, donc avec des accents, des cédilles ; ou même en chinois ou en arabe. L'inconvénient est que cela crée un frein pour internationaliser les scripts, en revanche les scripts sont rendus plus lisibles dans une langue donnée et par ailleurs, le code se rapproche de la langue parlée, rendant les scripts encore plus explicites.

Unicode

L'unicode est une façon de coder informatiquement les caractères pour qu'ils soient internationaux incluant les accents, les cédilles et les caractères de toutes les langues du monde (arabe, cyrillique, chinois, tibétain, etc.). Le codage se fait sur un plus grand nombre de bits que le codage ASCII (7 bits) fondé sur les caractères propres à la langue anglaise.

13.1 Unicode

L'unicode est donc utilisé dans les noms de variables et dans les chaînes de caractères. Ainsi les fonctions qui manipulent des chaînes de caractères multi-octets deviennent obsolètes puisque leur spécificité devient la règle

(`mb_strtolower`, `mb_strtoupper`, `mb_substr_count`, `mb_substr`, etc.). Ainsi PHP jusqu'à la version 6 manipule des données binaires et non des caractères, ce qui n'est plus le cas avec l'avènement de cette version 6.

```
class élucubrations {
var $délire="Fô s'qui fô!";
}
echo $délire ;
// Fô s'qui fô!
```

L'unicode, ce n'est pas seulement donner des noms bien de chez nous aux variables, c'est aussi économiser parfois du code. C'est en fait tout le contexte d'une langue qui est livré avec l'unicode :

- Les règles de tri d'une liste (`cote`, `coté`, `côté`, …) ;
- Les lettres accentuées mises en capitales accentuées (`Ç`, `À`, `È`, `Æ`, …) ;
- Les formats de date, d'heure ;
- Les formats de nombres, de nombres monétaires (`1.022,35 €`) ;
- Les noms de pays en français avec le code ISO ;
- Les noms de langues en français avec le code ISO ;
- Les noms de fichiers ;
- La transcription des chiffres en mots (*mille vingt-deux virgule trente-cinq euros*) ;
- Les symboles musicaux, mathématiques (, ∞, ≠, ≤, …) ;
- La translittération entre des caractères locaux vers des caractères latins et vice-versa ;
- Possibilité de découper les mots ou les lettres selon la langue ;

13.2 Le paramètre Register Globals n'est plus disponible

Ce paramètre avait fait le succès du PHP 3. Il permettait à des novices de faire du code simple pour récupérer les données d'un formulaire afin de les insérer dans une base MySQL.

Il s'est avéré dangereux car ces variables automatiques permettaient à quelqu'un de l'extérieur d'injecter du code puisque sa provenance n'était pas certaine.

13.3 Les Magic Quotes sont retirées

Les `magic_quotes` disparaissent. Ces paramètres permettaient d'ajouter des caractères d'échappement dans une requête SQL et ainsi éviter l'injection de code directement dans les formulaires. Il s'agissait juste d'une commodité pour les débutants afin de sécuriser directement leurs formulaires. Ces commodités ont permis au PHP d'avoir cette popularité mais ont fini par encombrer l'API.

Le site `Netlobo.com` propose une fonction pour sécuriser les formulaires et qui tient compte de toutes les versions de PHP et emploie la fonction PHP `mysql_real_escape_string()` qui a l'avantage de permettre d'échapper tous les caractères spéciaux.

```
function quote_smart($value)
{
    // Stripslashes
    if (get_magic_quotes_gpc()) {
//si le magic_quotes_gpc est activé
        $value = stripslashes($value);
    }
    // Quote if not integer
    if (!is_numeric($value)) {
        $value = "'" . mysql_real_escape_string($value) . "'";
    }
    return $value;
}
$nom= quote_smart($_POST['value']) ;
$sql= "INSERT INTO identite (nom) VALUES('$nom')";
```

Ce code permet d'éviter l'injection SQL dans vos formulaires.

13.4 Le Safe Mode n'existe plus

Le `safe mode` consistait à désactiver un certain nombre de fonctions PHP qui pouvaient causer des failles. Ce sont les fonctions de fichiers, les fonctions système ou des fonctions comme la fonction `mail()`. Pour un code mal écrit, elles peuvent être dangereuses.

13.5 Utiliser Freetype 2 et GD 2

Pour utiliser les *free fonts* et les graphiques sous PHP 6, il est nécessaire de se servir des fonctions propres à ces versions de Freetype et de GD car leur version 1 n'est plus dans PHP.

13.6 Plus de HTTP_*_VARS

Ce terme désigne tous les `HTTP_*_VARS` (`HTTP_POST_VARS`, `HTTP_GET_VARS`, …) qui désormais génèrent une erreur. Ceci a été décidé pour normaliser un peu plus PHP. Il est demandé maintenant d'utiliser exclusivement `$_POST`, `$_GET`, … .

13.7 Les extensions in et out

« Et les lauréats sont... : »

- Les modules *XMLReader* et *XMLWriter* qui permettent de lire ou d'écrire des flux XML ;
- Le module *fileinfo* qui permet de déterminer le type de codage d'un fichier.

Et les sortants sont :

- L'extension *ereg* et toutes les fonctions attenantes disparaissent du noyau et gagne le domaine des extensions *PECL*. La bibliothèque de manipulation de l'unicode *ICU* sera utilisée pour des expressions rationnelles avec les fonctions autour de *preg*.

13.8 Les ajouts

Des nombres entiers 64 bit

Les nombres entiers seront par défaut en 32 bits, excepté s'il est spécifié `int64`.

Goto

C'est une commande `goto` sous une autre forme, puisqu'il est maintenant possible dans une boucle de sortir de la boucle pour aller à une étiquette (*label*) à un autre endroit du code avec un `break` *etiquette*.

```
for ($i = 0; $i < 9; $i++)
  {
  if ($i==8) {
    break affiche;
  }
  echo "rien";
  affiche:
  echo "huit";
}
```

ifsetor()

`Ifsetor` est un raccourci. Ainsi ce code

```
 if(isset($_POST['âge']))
        {
                echo "Vous avez $_POST['âge'] ans";

        }
else {
        return "Vous n'avez pas indiqué votre âge";
}
```

peut être désormais remplacé par cette ligne

```
 $âge = ifsetor($_POST['âge'], "non mentionné");
```

Tableaux multi-dimensionnels avec foreach

Les tableaux multidimensionnels sont courants et PHP 6, là encore, simplifie la syntaxe. Là où il fallait deux `foreach` imbriqués il suffit maintenant d'une simple boucle :

```
$a = array(array(1, 2), array(3, 4)); // Tableau 2D
foreach( $a as $clef => $valeur) {
   list($a, $b) = $valeur; // affiche tout le contenu du tableau
   echo $b."<br />\n";
}
```

Ce code ci-dessus s'écrit maintenant avec PH 6 :

```
$a = array(array(1, 2), array(3, 4));
foreach( $a as $k => list($a, $b)) {
   echo $b."<br />\n";
}
```

Espaces de nom

Les espaces de noms consistent en une classification de classes d'objets. Cela permet d'appeler toutes les classes d'un espace de nom en une seule fois ou d'éviter la collision entre des classes qui auraient le même nom.

```
namespace backoffice {
        class article {
        }
}
```

APC entre au cœur

Le cache d'opcode APC devient un standard pour PHP 6, ce qui accélère l'exécution du code.

opcode

L'opcode est du code prédigéré. C'est l'étape intermédiaire entre un script et le processeur PHP.

Au lieu de répéter les même processus de transformation du code lisible, APC stocke les codes machine.

<%, <?php, <? ?

Les tags <% %> encadrant le script php et ressemblant au langage ASP n'existent plus. Les deux autres sont toujours de mise.

14

Annexe

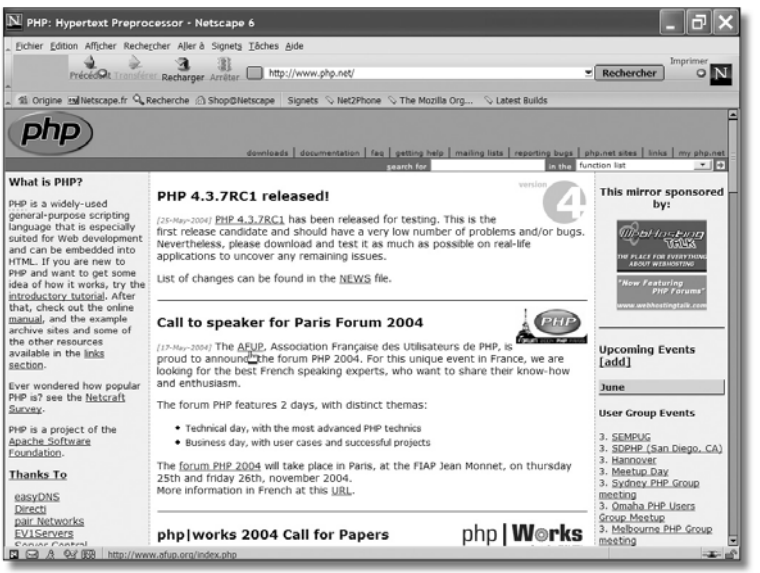
Voici quelques références web utiles. La plupart des sites référencés dans l'annexe sont en français et la majorité des logiciels fonctionnent sous Windows. Nous avons indiqué quand ce n'était pas le cas.

14.1 PHP

PHP Hypertext Preprocessor

Le site central et officiel de PHP. Rédigé en anglais, il contient les dernières nouvelles, les projets, les versions et la documentation annotée. Des versions dans toutes les langues existent.

`www.php.net`

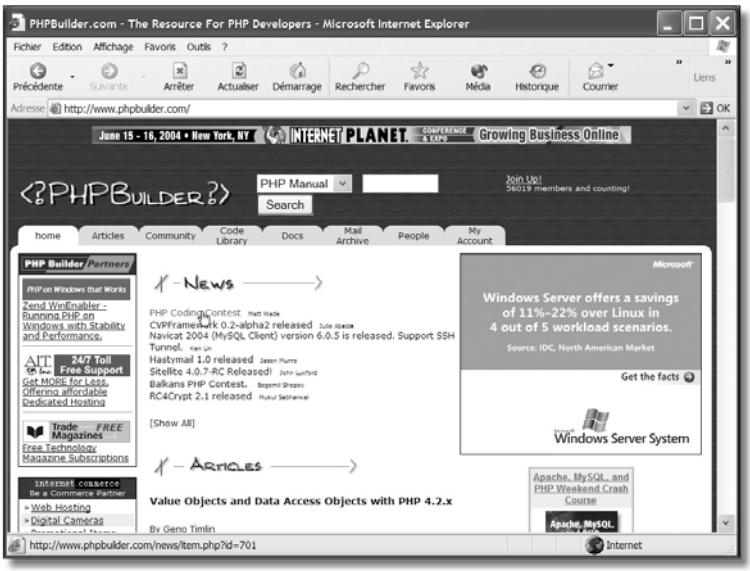


▲ Figure 14.1 : *www.php.net*

PHP Builder, the Resource for PHP developers

Un site en anglais, très professionnel, qui présente des articles de fond, des annotations très utiles pour accompagner la documentation, une bibliothèque de codes, etc.

`www.phpbuilder.com`



▲ Figure 14.2 : *www.phpbuilder.com*

PHP Index, la Passerelle française des technologies PHP

Un des meilleurs sites en français. Vous y trouverez des astuces, des annonces, des liens et des références… C'est un site très complet et professionnel.

`www.phpindex.com`

▲ Figure 14.3 : *www.phpindex.com*

Nexen

Les deux webmestres de ce site ont traduit la documentation PHP et *MySQL* pour l'offrir aux internautes. Sur le site, elle apparaît en français et annotée. Le site contient des astuces, des forums et les dernières nouvelles.

`www.nexen.net`

▲ Figure 14.4 : *www.nexen.net*

PHP Facile

Un site destiné aux débutants en PHP mais pas exclusivement. Il contient des cours de PHP et *MySQL*, un manuel et répertorie les erreurs fréquentes.

`www.toutestfacile.com/phpinit.php?tef_site=php`

▲ Figure 14.5 : *www.toutestfacile.com*

ASP-PHP.NET

Ce site est le résultat d'un travail approfondi autour du PHP, ASP et .NET, sans oublier le XML et d'autres langages. De nombreux scripts et tuteuriels où l'ASP et le PHP sont parfois comparés.

`www.asp-php.net`

▲ Figure 14.6 : *www.asp-php.net*

PHP France

Un site avec beaucoup d'astuces, des tuteuriels, des annonces et des morceaux de code.

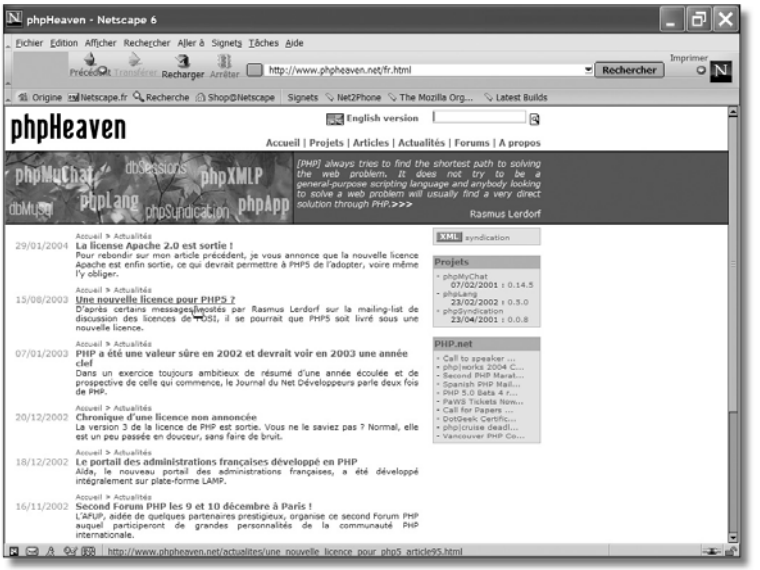
`www.phpfrance.com`

▲ Figure 14.7 : *www.phpfrance.com*

PHP Heaven

Ce site répertorie des projets professionnels autour de PHP, comme Pear, et contient des articles.

`www.phpheaven.net`



▲ Figure 14.8 : *www.phpheaven.net*

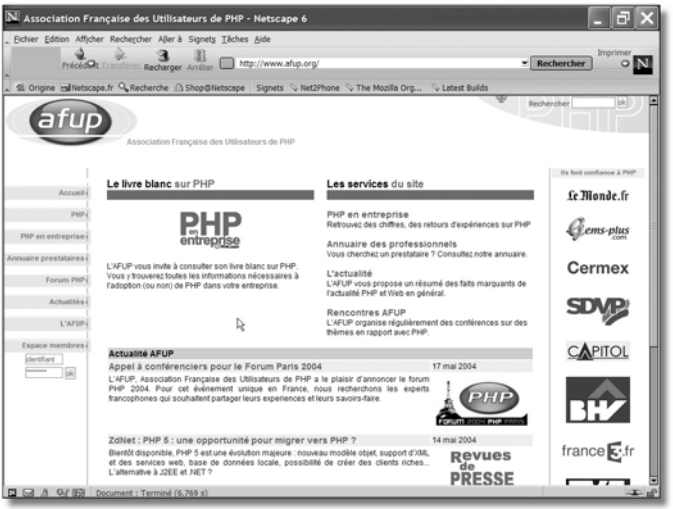
AFUP

L'AFUP, Association française des utilisateurs de PHP, est une association dont le but est de promouvoir le langage PHP auprès des professionnels et de participer à son développement.

`www.afup.org` (voir fig. 14.9)

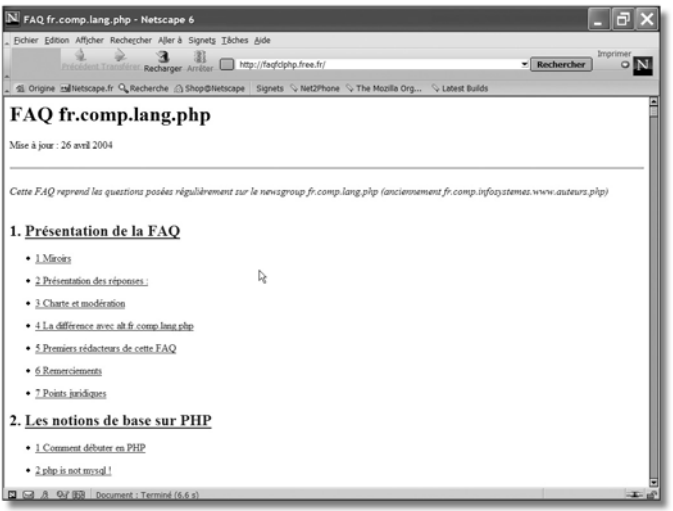
Le FAQ du groupe de news faqfciwap (`fr.comp.lang.php`).

Le site indispensable pour tout développeur PHP, une encyclopédie du langage PHP.



▲ Figure 14.9 : *www.afup.org*

`http://faqfclphp.free.fr`



▲ Figure 14.10 : *http://faqfclphp.free.fr*

14.2 MySQL

MySQL

Le site officiel de *MySQL* avec les nouvelles, les documentations et les dernières versions. Site en anglais.

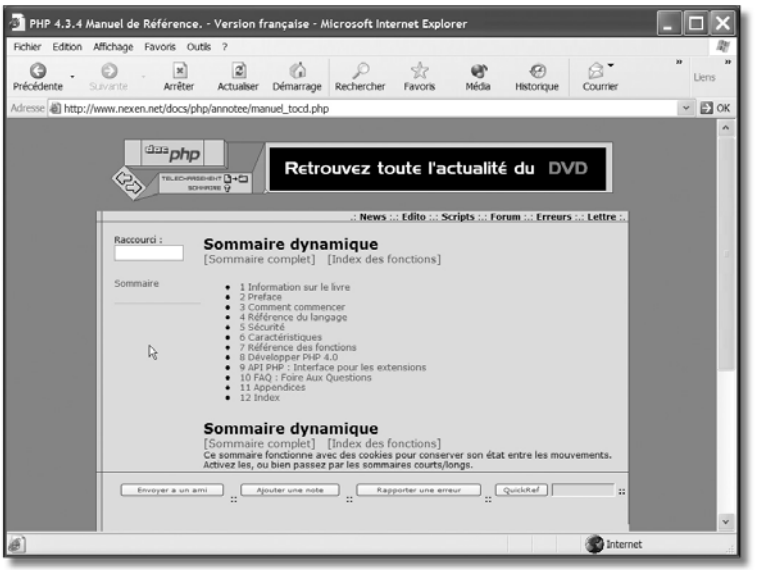
`www.mysql.com`

▲ Figure 14.11 : *www.mysql.com*

Nexen

Toute la documentation *MySQL* annotée en français.

`www.nexen.net`



▲ Figure 14.12 : *www.nexen.net*

14.3 Autres

Apache

Les dernières versions du célèbre serveur et tous les projets de la communauté du Libre autour de ce serveur. Site en anglais.

`www.apache.org`

▲ Figure 14.14 : *www.apache.org*

Easyphp

Installez-le sur votre machine. Il inclut PHP, *MySQL* et Apache pour Windows.

`www.easyphp.org`

▲ Figure 14.14 : *www.easyphp.org*

14.4 Outils

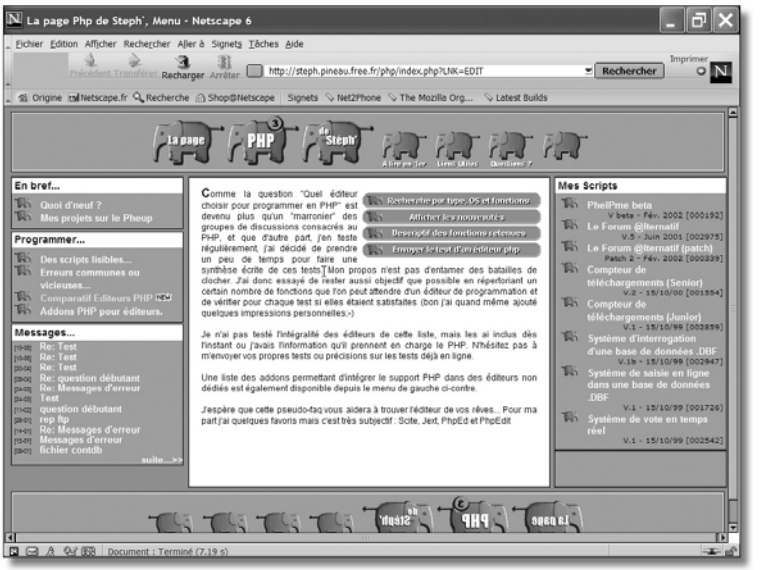
Cette petite liste ne se veut pas exhaustive mais constitue un solide bagage.

Éditeurs

Liste des éditeurs PHP

Une liste sous forme de tableau avec les caractéristiques de chaque éditeur de code PHP.

`http://steph.pineau.free.fr/php/index.php ?LNK=EDIT`

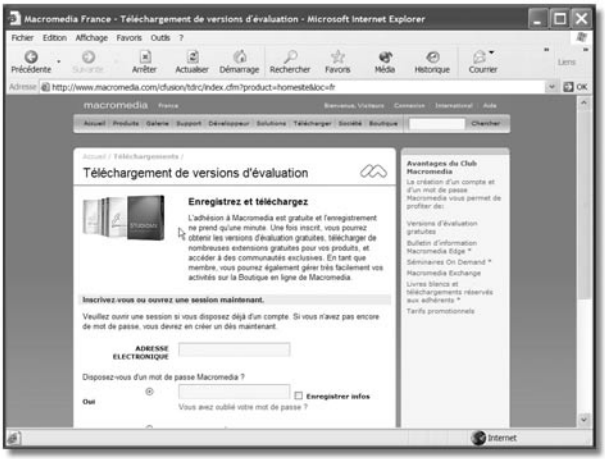


▲ Figure 14.15 : *http://steph.pineau.free.fr*

Homesite de Macromedia

Un des meilleurs éditeurs de PHP payants.

`www.macromedia.com/fr/software/homesite/trial/`



▲ Figure 14.16 : *www.macromedia.com*

PhpCoder

Un très bon éditeur de PHP gratuit, presque l'équivalent de Homesite.

`www.phpide.de`

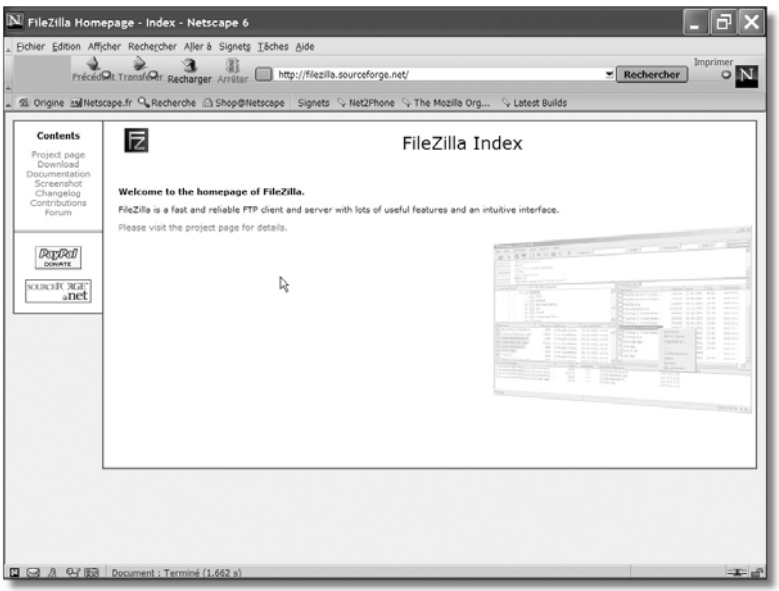
▲ Figure 14.17 : *www.phpide.de*

FTP

Filezilla

C'est un logiciel FTP open source et gratuit.

`http://filezilla.sourceforge.net`



▲ Figure 14.18 : *http://filezilla.sourceforge.net*

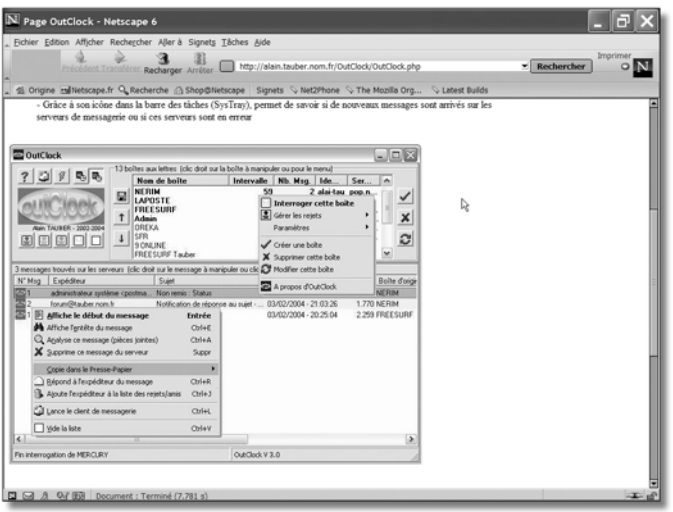
Fetch

Le client FTP le plus courant pour Macintosh.

`http://fetchsoftworks.com`

▲ Figure 14.19 : *http://fetchsoftworks.com*

Personnels



▲ Figure 14.20 : *http://alain.tauber.nom.fr*

Outclock

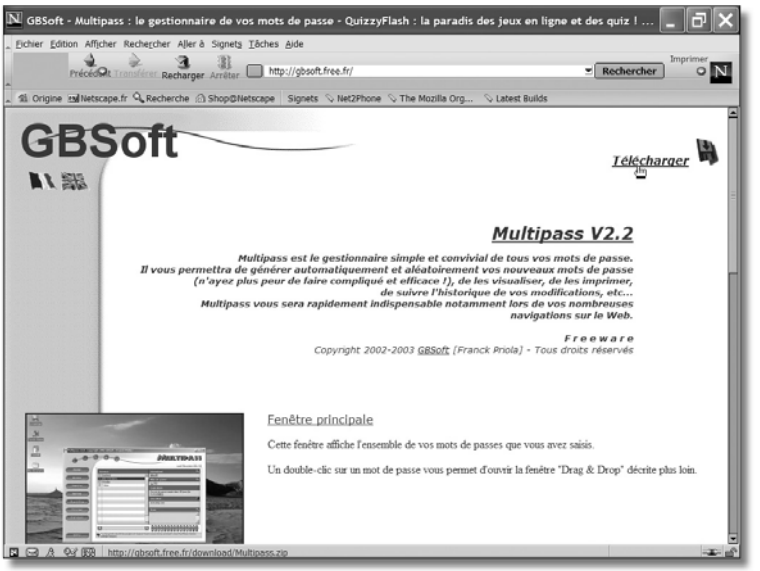
Outclock est un logiciel gratuit en français qui vous permet de lire vos courriels sans les télécharger sur votre disque dur. Il vous permet de rejeter un spam ou à détruire un virus. C'est pratique aussi pour lire son courrier personnel au bureau sans le mélanger avec le courrier professionnel.

`http://alain.tauber.nom.fr/Outclock/Outclock.php`

Multipass

Multipass est un coffre-fort à mots de passe. Indispensable pour vous y retrouver dans vos nombreux futurs sites.

`http://gbsoft.free.fr`



▲ Figure 14.21 : *http://gbsoft.free.fr*

Crypter un mot de passe

Une page qui vous permettra de crypter le mot de passe que vous insérerez dans le fichier *.htpasswd*.

`www.euronet.nl/~arnow/htpasswd`

Un antivirus gratuit

Un site exhaustif sur la sécurité qui propose un antivirus gratuit et efficace.

`www.secuser.com/antivirus`

Un outil pour les URL interminables

Ce site crée une URL courte qui renvoie à l'adresse longue que vous avez entrée. Cela évite d'avoir une URL coupée dans un courriel, surtout quand vous avez de nombreuses variables à la suite comme dans une librairie en ligne.

`http://www.tinyurl.com`

14.5 Applications en PHP

Hotscripts

Une importante bibliothèque d'applications en anglais

`www.hotscripts.com/PHP/Scripts_and_Programs`

SourceForge

Une forge pour les outils et les projets de la communauté du Libre. Site en anglais.

`http://sourceforge.net`

PHPscripts

Plus de 500 applications en PHP sur tous les thèmes.

`http://www.phpscripts-fr.net`

PHPbb

Un forum en PHP pour votre site.

`www.phpbb-fr.com/`

Phorum

Un autre forum en PHP. Site en anglais.

`www.phorum.org`

Atomz

C'est un moteur de recherche personnel et gratuit pour votre site, en français. Vous personnalisez votre page de résultats aux couleurs de votre site. Des statistiques de recherche vous sont fournies. L'indexation sera manuelle à partir du site.

`www.x-recherche.com/services.html`

PHPMyAdmin

Téléchargez l'outil d'administration le plus répandu pour *MySQL*.

`www.phpmyadmin.net/home_page`

PhpMyChat

Une causerie en PHP créée par Nicolas Hoizey.

`www.phpheaven.net/projects/phpMyChat`

14.6 Les portails

Typo3

Typo 3 est un CMS (pour `Content Management System`, système de gestion de contenu. Ce système permet de gérer un site collectivement et simplement avec des possibilités de personnalisation de l'interface graphique.

`www.typo3.fr`

Spip

Une application très bien conçue avec des zones réservées. Idéale pour créer un magazine en ligne contenant de nombreuses contributions.

`www.spip.net/`

14.7 Hébergement

Pour l'hébergement, vous avez le choix entre les hébergeurs gratuits avec ou sans publicité et les sites payants qui vous garantissent plus de bande passante, donc un accès plus rapide avec votre propre nom de domaine. Les sites gratuits se raréfient, absorbés par de grands groupes. Vous pouvez aussi vous tourner vers votre fournisseur d'accès internet.

Gratuit

ifrance

Un des survivants des hébergeurs gratuits. L'offre comprend PHP4 et *MySQL*. Une bannière de publicité jaillit à chaque chargement de page.

`www.ifrance.com/heberg/accueil/`

WebDynamit

WebDynamit.net est une communauté de webmasters bénévoles qui offre des services Web gratuits depuis mai 2002. Aujourd'hui, plus de 1600 sites sont hébergés sur leurs serveurs sanas publicité ! N'hésitez pas à les aider, ce n'est pas si courant.

`www.webdynamit.net`

Free

Votre site est hébergé sans publicité mais vous ne disposez pas de la fonction `mail()` et vous disposez de PHP version 4.

`www.free.fr`

La Poste

Pour avoir ses pages perso, il suffit de souscrire à l'accès libre (vous n'êtes pas obligé de l'utiliser). La configuration comprend PHP4, MySQL, GD et 50 Mo.

`http://offres.accesinternet.laposte.net/`

Payant

ABC Hébergement

Le moteur de recherche d'hébergeurs payants. C'est un site incontournable pour celui qui cherche. Vous pouvez rechercher parmi les hébergeurs qui proposent un serveur mutualisé (vous partagez votre serveur avec d'autres clients, en ayant votre nom de domaine) ou un serveur dédié (une machine pour vous), ou qui hébergent des machines (vous leur confiez votre propre serveur). Comparez les prix et les offres.

`www.abchebergement.com`

Ouvaton les amis

Un OVNI dans l'hébergement. C'est une coopérative. L'hébergement coûte pour l'instant environ 12 €/an pour une grande capacité.

`http://ouvaton.coop/`

Nexen

Ils ont des offres payantes intéressantes avec la version 4 de PHP.

`www.nexenservices.com`

Noms de domaine

Gandi

Gandi gère les noms de domaines. Leur foire aux questions (FAQ) est très complète pour que vous compreniez comment procéder avec un nom de domaine. Leurs prix sont intéressants et l'interface vous aide à gérer tous vos noms de domaines.

`www.gandi.net`

14.8 Référencement

Référencement 2000

Cette offre gratuite vous référence auprès de la plupart des moteurs de recherche et annuaires de sites. Cela ne vous dispense pas de vous faire connaître par d'autres moyens.

`www.referencement-2000.com/refgratuit.html`

14.9 HTML et Javascript

All HTML

Un site très complet sur la plupart des langages. Nous vous le conseillons pour HTML et Javascript.

`www.allhtml.com`

Le Journal du net

Le Journal du net dans sa partie pour les développeurs, évoque plusieurs langages dont le PHP. Des astuces et des tuteuriels sont proposés.

`http://developpeur.journaldunet.com/`

Comment ça marche

Un site de vulgarisation sur l'informatique, les différents langages, la sécurité, le web, les bases de données, la législation, les réseaux, les systèmes d'exploitation… Excellent site avec des articles très clairs et un glossaire des termes informatiques.

`www.commentcamarche.net`

14.10 Les groupes de news

- PHP, fclp (`fr.comp.lang.php`) ;
- PHP, ap (`alt.php`) ;
- SGBD, fcas (`fr.comp.applications.sgbd`) ;
- SGBD, aps (`alt.php.sql`) ;

  SERVEUR, fciws (`fr.comp.infosystemes.www.serveurs`).
  Un site pour consulter tous ces newsgroups

`www.1001newsgroups.com`

Pour écrire à l'auteur de ce livre

L'adresse courriel de l'auteur est `livrePHP@carfantan.com` et son site est `http://jean.carfantan.com`. Toute suggestion ou remarque sera la bienvenue.

15

Index

!

!= ou !== 69
= 49, 69, 69, 82
== 69
=== 69
? 38, 184
..... 181
#. 308
& 68
* 115
- 303
/ 308
\n 60, 131
\r 60
\t 60
\$ 60, 302, 308
$DOCUMENT_ROOT 177
$PHP_AUTH_PW 173, 175
$PHP_AUTH_USER 173, 175
$PHP_SELF 146, 193
$REQUEST_URI 143
$SCRIPT_FILENAME 255
$SCRIPT_NAME 256
$_GET 383
$_POST 383
.htaccess 190-191
.htpasswd 190-191

 329
<title> 319

A

Abs() 265
Accolades 62
ADD 113
Addition 67
AddSlashes() 86
Adresses e-mail
  obsolètes 196
  périmées 197
Affectation 67
AFUP 393
Agenda 223
Alert() 135, 139-140
Algorithme 77
ALTER 127
  ALTER TABLE 113
AND 68, 72, 125, 153
Annuaire de liens 276, 377
APC 385
Apostrophe 57
Arbre de possibilités 366
Architecture 364
Archive 313
Array() 210
Array_count_values() 300
Array_diff() 211, 264
Array_flip() 300
Array_intersect() 211
Array_keys() 210
Array_merge() 210
Array_merge_recursive() 211
Array_pop() 211
Array_push() 211
Array_shift() 211
Array_slice() 211
Array_splice() 211
Array_unshift() 211
Array_values() 211
Array_walk() 211
Arrondi 265
Arsort() 214
AS 154-155
Asort() 214
Attributs 161, 324
Authentification 172
Autocomplétion 353
AUTO_INCREMENT 110
AVG() 152

B

Barre oblique inversée 58
Base de données
  sauvegarder 118
Basename() 256
BETWEEN 153
BIGINT 117
Bissextiles 81
Bits 68
BLOB 116
Boolean 55
Booléens 56
Boucles 77

Boutique 258, 333
Boutique électronique 332
Bouton radio 136, 233
Break 84, 229, 368
Brevet logiciel 20

C

C++ 21
Caddie 332
Cadriciels 345
CakePHP 346, 355
Calendrier 219
Caractère 309
  d'échappement 58
  d'espace 309
  de mot 309
  de non mot 309
  de nouvelle ligne 309
  joker 316
  non numérique 309
  numérique 309
Cardinalité 305
Case à cocher 136
CDATA 326
Ceil() 243, 265
Cellules
  fusionner 35
CGI 201
Chaînes de caractères 57, 86
Champ
  Propriétés 137
  caché 184
CHANGE 114
CHAR 116
Chariot électronique 258
Chiffrer.inc.php 195
Chmod 252
Chmod() 248, 251
Chop() 90
Classe 161
Clé
  étrangère 114
  primaire 112
  secondaire 112
Client 366
Club
  conception 108
CMS 405
Code hexadécimal 31
Commentaires 51
Commit 334
Communauté du Libre 18
Communication 373
Compact() 211
Compilateur 23
Compteurs 254
Concours 247
Conf.ini 351
Configuration 51
Connexion 119
Connexion.inc.php 120
Constantes 60
Constructeur 162, 168
Continue 85
Contrôle d'erreur 70
Contrôleur 347, 349
Cookie 179, 181
  effacer 180
Copy() 249
Count() 100, 152, 155, 212, 339
Courriel 375
Courriel en HTML 193
CREATE 111, 152
Critères de nommage 54
CRUD 350
Cryptage 191, 195
CSS 329
CSV 335

D

Date 88, 113, 117, 312
Date() 218
Dates 218
DATETIME 118
Date_nombre() 89, 307
Débogage 367
Décalage à droite ou à gauche 68
DECIMAL 116-117
Décrypter 195
Default 77, 112
Define() 60
DELETE 130, 152-153, 185
Delete() 249
Délimiteurs 315-316

Demission.php 187, 196
Dirname() 256
DISTINCT 153, 282
Division 67
Do while 79
Document Object Model 134
Documentation
  MySQL 390
  PHP 390
DOUBLE 117
Doublons 112
Dreamweaver 44, 369
Drupal 346, 360
DTD 322, 324, 326

E

E-mail 130
  format HTML 131
  format texte 131
  formater un courriel 35
Each 83
Each() 212
EasyPHP 45
Échappement 60, 124
Echo 58
Éditeurs PHP 398
Élaboration d'une application 365
Elements[] 137, 323
Else 75
Elseif 76
Emails_perimes 196
Émoticônes 375
Empty() 143
En-têtes d'un courriel en HTML 132
End() 213
Entiers 55
Envoi en nombre 192
Envoi() 138, 140
Ereg() 302
Eregi() 302
Eregi_replace() 303
Ereg_replace() 302
Erreur 70, 366
  de requête SQL 369
Espace de nommage 323, 328, 385
Espace insécable 222
Exit 120, 368
Expat 324
Explode() 90
Expression
  rationnelle 301
  rationnelles compatibles Perl 316
  régulière 301
Extract() 212

F

False 55-56
Fclose() 249
Feof() 249
Fetch 47, 400
Fgetcsv() 249
Fgets() 249
Fgetss() 249
Fibonacci 102
Fichier
  données 108
  inclus 120
Fields.ini 352
File() 249
Fileatime() 251
Filectime() 251
Fileinfo 383
Filemtime 368
Filemtime() 251
Fileperms() 251
Filesize() 251
Filetype() 251
Filezilla 400
File_exists() 251
Float 55-56, 117
Floor() 243, 265
Focus() 139
Fonctions 85, 97
  imbriquées 101
  nombre 13
  récursives 102
Fopen() 249
For 80
Force Brute 176
Foreach 82, 214, 262
FORM 134
Formulaire 83, 119, 121, 184-185
  champ caché 148
  checkbox 39
  file 40
  hidden 39

image ... 40
INPUT ... 39
nom des champs ... 122
option ... 40
password ... 39
radio ... 39
select ... 40
sélection multiple ... 141
submit ... 39
text ... 39
textarea ... 40
wrap ... 40
Fpassthru() ... 250
Fputs() ... 250
Frameworks ... 345
Fread() ... 250
Freetype ... 383
Fscanf() ... 250, 253
Fseek() ... 250
Ftell() ... 250
Fwrite() ... 250

G

GD ... 383
General Public License ... 20
Getc() ... 249
Get_html_translation_table() ... 300
Get_meta_tags() ... 298, 300
Glossaire ... 408
Google ... 377
Goto ... 384
GROUP BY ... 153
Groupes de news ... 408
Guillemets ... 50, 57
doubles ... 59
simples ... 57

H

Header() ... 173
Hébergement ... 47, 405
Hébergeur ... 46, 192
Héritages ... 163
Hidden ... 150, 184
Histoire ... 21
HomeSite ... 44
HTML ... 26, 407
alink ... 35
cadres ... 40-41
commentaires ... 51
formulaire ... 38
images ... 32
liens ... 33
link ... 35
listes ... 31
noframes ... 42
NORESIZE ... 42
notions ... 28
page index ... 29
polices de caractères ... 30
SCROLLING ... 42
tableaux ... 35
titre de paragraphe ... 31
vlink ... 35
Htmlentities() ... 86
Htpasswd ... 403
HTTP ... 176, 201
en-tête ... 201
Https ... 182
HTTP_GET_VARS ... 383
HTTP_POST_VARS ... 383
Humour ... 374

I

If ... 74
Ifsetor ... 384
Images
balise ... 33
bordure ... 34
gif ... 32
jpeg ... 32
texte ... 33
transparence ... 32
Implode() ... 90
Include ... 123, 128, 143
Index ... 183
Indexation ... 112, 297
Injection SQL ... 382

Inscription au club 119
Inscription_p1.php 177
INSERT 152
Instance 161
Int 55, 117
Integer 55
In_array() 212
Isset() 143
Is_dir() 251
Is_executable() 251
Is_file() 251
Is_linkinfo() 251
Is_readable() 251
Is_writeable() 251

J

Java 21
Javascript 134, 407
  confirm() 186
  événements 137
  fonction 186
  fonctions 134, 138
  length 135, 139
  méthodes 136
  objets 134
  parseInt() 135
  tableaux 135
  variable globale 138
  variables 134
Jointure 115
Jointure externe 197
Joomla 346

K

Key() 212
Krsort() 214
Ksort() 214

L

Langage
  bas niveau 23
  C 23
  haut niveau 23
  naturel 21
  objet 21
LAST_INSERT_ID() 281
Length 135
Lettres accentuées 305
Libre 19
Lien
  affichage 283
  e-mail 34
  fenêtre 34
  TARGET 43
  vérification 293
  vers un paragraphe 34
LIKE 154-155
LIMIT 154
Limite
  de mot 309
  de non mot 309
Linus Thorvalds 20
Linux 20
List 83
List() 212
Logiciel
  d'édition de code 43
  FTP 46
  libre 20
Luelists.ini 353

M

Magazine 265
Magic Quotes 382
Magic_quotes_gpc 87
Magic_quotes_runtime 87
Mail() 130, 192, 366
Mailto 317
Mambots 359
Marqueurs 146-147
Match 330
Max() 265
MAX_FILE_SIZE 268
Mb_strtolower 381

Mb_strtoupper 381
Mb_substr 381
Mb_substr_count 381
MEDIUMINT 117
Mémoire 156
Menu déroulant 114, 122, 136
Meta 377
Méthode
  GET 38
Méthodes 161
Méthologie 347
Min() 265
Mini-forum 236
Mkdir() 250
Mktime() 218
Modèle 265, 330, 347-348
Modèle/Vue/Contrôleur 346-347
Modéliser 162
Modificateur 315
Modification de l'email 199
Modulo 67, 81, 221
Mot de passe 176
Moteur de recherche 295, 377
Multiplication 67
MVC 347
MySQL 334
  fonctions 151
  histoire 108
Mysql_connect() 120, 133
Mysql_create_db() 133
Mysql_db_query() 124, 133
Mysql_fetch_array() 133
Mysql_fetch_array(). 124
Mysql_insert_id() 264, 281
Mysql_num_rows() 133
Mysql_real_escape_string 382

N

Name 122
Namespace 385
Nétiquette 375
Newsgroups 408
Next() 213
Nl2br() 90, 131
NO 68
Nombres
  à virgule flottante 56
  pseudo-aléatoires 96
Nombre_date() 89, 306
Noms
  de domaines. 407
  homogénéité. 111
Non avide 316
NOT 72, 153
NULL 56, 112

O

Objet 161
OnBlur 137
OnChange 137
OnClick 137
OnFocus 137
OnLoad 199
OnSelect 137
OnSubmit 137-138
Open Source .....
  18
Opérateurs
  chaîne de caractères 73
  de bits 68
  logiques. 72
  priorité. 72
  ternaire 69
  unaires 71
OR 68, 72, 125, 153
ORDER BY 125, 154
Oubli.php 146, 151
Ouvrir_vasistas() 151

P

Page d'accueil 41
Pagination 340
Paramètres 339
Password 127
PCDATA 326
Pear 393

PECL 383
Perl 21, 307
Permissions 252
PHP
  balises 125
  commentaires 51
  communauté 19
  extensions 13
  fonctions MySQL 133
  GTK 24
  sigle 18
  vérification des données d'un formulaire 141
PHP 6 379
PHP Coder 43
PHPinfo 51
Phpinfo() 87
PHPmyAdmin 116
PHPNuke 372
PHP_AUTH_PW 180
PHP_AUTH_USER 180
PI 95
Plone 348
Point d'interrogation 184
Pointeur 213
Polymorphisme 166
Portabilité 20
Pos() 213
PRE 59
Précédence 72
Preg 383
Preg_grep() 308
Preg_match() 298, 308
Preg_match_all() 308
Preg_replace() 281, 308
Preg_split() 308
Prev() 213
Privilèges 168
Prix 116
Procédures stockées 156
Propriétés 161
Prototype 161

Q

Quote_smart 382

R

Racine 334
RACINEWWW 177
RAND() 152
Range() 211
Readfile() 250
Readlink() 251
Référencement 377, 407
Référenceurs 377
Register Globals 381
Relationships.ini 352
Rename() 250
Requête 126
  imbriquées 157
Reset 39
Reset() 213
Return 368
Rewind() 250
Richard Stallman 20
RLIKE 154
Robots.txt 294
Rollback 334
Round() 265
Rsort() 214
RSS 341
Ruby on Rails 355, 361

S

Safe Mode 382
Screen.height 193
Screen.width 193
SELECT 115, 125, 152
Selected 123
SelectedIndex 136
Serveur
  Apache
  dédié 406
  mutualisé 406
  statistiques 23
Session_register() 149, 151, 183
Session_start() 183, 188
Setcookie() 182

Setlocale() 94
SGML 322
Shuffle() 214
Signe d'échappement 57
SimpleXML 340
Size 127
Sizeof() 212
SMALLINT 117
Smarty 348
Sondage 240
Sort() 214
Soustraction 67
Soutien 374
SPIP 372
Split() 303
Sprintf() 253
SQL 151
SQL 2003 156
SQLite 156, 334
Sql_regcase() 303
Sscanf() 253
Stat() 251
Statistiques 254
Stats.inc.php 256
Strftime() 94
String 55, 57, 86
Stripslashes 382
StripSlashes() 86-87
Strip_tags() 86, 91
Strlen() 86-87
Strtolower() 91, 149
Strtoupper() 91
Structures de contrôle 74
Str_replace() 91, 100
SUBMIT 134
Substr() 86-87
Suivi.php 184
Sun 21
Surcharger 165
Switch 77
Syndication 341
Système
  de fichiers 249
  de gestion de contenu 405

T

Table
  création 109
  transitoire 295
Tableau 63
  à index numérique 63
  colonnes 35
  de bord (membre) 184
  de hachage 63-64
  multi-dimensionnels 384
  multidimensionnels 63, 65
  rangs 35
Tables d'adressage associatif 64
Templates 265, 330, 359
Test 76
Tests 74, 366
TEXT 116, 224
Textarea 233
Thatware 372
TIME 118
Time() 180
TIMESTAMP 113, 117, 218, 282
TINYINT 117
<title> 319
Touch() 251
Traitement_modif_profil.php 189
Tri
  associatif 215
  simple 215
Trim() 86-87, 100
True 55, 57
Type 89, 307

U

Uasort() 214
Ucfirst() 91
Ucwords() 91
Uksort() 214
Umask() 251-252
Unicode 380
UNIQUE 112
Unix 20
Unlink() 250
UPDATE 152, 185
URL 38
Urlencode() 38, 184
Usort() 214

V

Value 127
VARCHAR 116
Variable 339
  de session 149
  dynamique 62
  extrapolation 62
  nom 54
  prédéfinies 145
  scalaires 54
  transtypage 55
  type 55
Vérification 128, 130
Verif_champs() 138-139
Verif_email () 139, 138
Votre_email.php 199
Vues 347

W

WHERE 125, 153
While 78
Window 135
Window.close() 150
Window.open() 151
WWW 177

X

Xataface 346, 348-349
XHTML 329
XML 322
XMLReader 383
XMLWriter 383
XOR 68, 72
XSLT 322, 328, 336
Xslt_errno() 336
Xslt_error() 336

CSS

1 Introduction 429

1.1. Évolution du HTML 431
1.2. Le tandem CSS et XHTML 432
1.3. Les navigateurs 432
1.4. Le concept de feuille de style 434
1.5. De la feuille de style à la feuille de style en cascade 434
En ligne 435
En-tête HTML 435
Fichier CSS 436
1.6. La notion de mise en page 436
Le lien hypertexte et son usage 438
L'harmonie des couleurs 439
La répartition des blocs 440
La lisibilité du texte 441
1.7. Les cascades de style 442
1.8. Le XHTML 444
1.9. La structure d'une page 445
Écrire pour le Web 445
Hiérarchisation de l'information 447
Communication plus simple entre les machines 448
1.10. Conclusion 449

2 La réflexion 451

2.1. Les types de sites 452
Les sites vitrine ou sites plaquette 452
Les sites catalogue 453
Les sites d'information 454
Les sites marchands 455
Les sites institutionnels 456
Les sites personnels 457
Les sites communautaires 458
Les sites intranet et les sites extranet 459
2.2. Quelques règles générales 459
2.3. Le site 459
2.4. Le public 460
2.5. La carte du site 462
2.6. Identifier le contenu sémantique 463
2.7. La navigation et la disposition des pages 464
2.8. Choisir les outils 464
TopStyle Pro 464
Master Styler 465
ColorZilla 466
Colourificator 466
EclipsePalette 467

Color Schemer 467
IconArt 468
Web Developer 469
2.9. Créer l'image du site 470
2.10. Bâtir le site 471
Le code XHTML 471
La feuille de style 471
La fonctionnalité du site 478
Le bon référencement 479
2.11. Internet n'a rien inventé 485

3 Les règles de mise en page 487

3.1. Le texte 488
Les différents types de polices 490
Taille absolue ou relative 491
Un exemple de taille relative 494
Les paragraphes 495
3.2. Les illustrations 498
3.3. La disposition 500
3.4. La couleur 502
Notation hexadécimale 505
Notation par mot-clé 506
Notation en pourcentage 507
Notation décimale 508
Notation en référence à l'interface de l'utilisateur 508
Les couleurs et images d'arrière-plan 510
3.5. La pagination 512
Faciliter l'identification de la page 513
Reporter les éléments de navigation 513
Garder la cohérence 514

4 La réalisation du site 515

4.1. La page d'accueil 516
4.2. Le site 519
4.3. Le XHTML 520
La balise fermante 521
L'imbrication des balises 523
Les dénominations en minuscules 524
Les propriétés définies 524
Le standard 525
4.4. Les CSS 526
Les sélecteurs 526
Les sélecteurs de type 527
Le sélecteur universel 527
Les sélecteurs descendants 528

Les classes ... 529
Les identifiants ... 530
L'héritage ... 531
Les sélecteurs d'enfant ... 532
Les sélecteurs d'enfant adjacents ... 534
Les sélecteurs d'attribut ... 534
Les déclarations multiples ... 537
Le regroupement ... 537
4.5. L'habillage de la page ... 538
4.6. Relooker des formulaires ... 542
Soumettre un formulaire ... 542
Un peu de couleur sur un menu déroulant ... 545
Les listes ... 546
Les pseudo-classes ... 548
Les pseudo-éléments ... 552
Les filtres ... 555
Selon le type de média ... 557
Bidouillages et défaillances autour d'Internet Explorer ... 561
4.7. Conclusion ... 568

5 Études de terrain ... 569

5.1. Quelques sites ... 570
Un site communicant ... 570
Un site appétissant ... 573
Un site design ... 579
Clair et corporate ... 589
5.2. Atelier d'artiste ... 604
5.3. Un site fédérateur ... 606
5.4. Construction du site ... 616
Réflexions ... 617
Types de boîtes ... 618
Position des boîtes ... 619
Une page à trois colonnes avec la position relative ... 626
Une page à trois colonnes avec la position flottante ... 632
La construction du site Écoute S'il Pleut ... 636

6 Les règles d'accession à la propriété ... 653

6.1. Le doctype ... 656
6.2. La langue ... 657
6.3. Les titres ... 658
6.4. Les liens ... 658
6.5. Les éléments de formulaires ... 659
6.6. Les tableaux ... 661
6.7. Les listes ... 664
6.8. Les images ... 667

6.9. Les abréviations 668
6.10. Le charset ou tableau de caractères 668
6.11. La navigation 669
Une navigation homogène 670
Les touches d'accès 671
6.12. L'ordre de lecture 673
6.13. Les aides 674
La page d'aide 674
Les aides invisibles de navigation 675
6.14. La page nue 676
6.15. Conclusion 676

7 Le ravalement de façade 677

7.1. Quand opérer une refonte d'un site ? 678
7.2. Où commencer ? 679
7.3. La question des tableaux 681
7.4. La refonte du site photo-de-classe.com 682
L'évolution du site 687

8 Vers la CSS3 701

8.1. Placer des boîtes 702
8.2. Autres caractéristiques 707
Les couleurs 707
Le son 708
Les polices 709
Les fonds 709
8.3. Conclusion 709

9 Les CSS se marient bien 711

9.1. Le JavaScript 712
JavaScript, le cerveau du navigateur 716
Quelques amuse-gueules 717
Récapitulation 722
Les différents navigateurs et le JavaScript 733
Quelques outils complémentaires 741
Les cookies 745
Le modèle objet de document (DOM) et AJAX 755
9.2. Le PHP 769
Ma page 770
Les variables scalaires 773
Créer un aiguillage de feuilles de style 776
9.3. Le XML 777
Qu'est-ce que le XML ? 778

La différence entre le XSLT et les CSS 779
Le fichier XML 781
La DTD 785
Le XSLT 787
Le fichier PHP 791
9.4. Conclusion 792

10 Références 793

10.1. CSS 794
10.2. XHTML 795
10.3. Accessibilité 796

11 Index 797

Préface

Cet ouvrage est destiné à tous les publics. La seule condition est de connaître modérément le HTML, qui est le langage vecteur d'Internet. Les CSS (*Cascading Style Sheets*), ou feuilles de style en cascade, sont l'essence même d'un site et touchent tous les domaines. Elles sont l'aboutissement d'un long processus de réflexion autour de la mise en page, l'utilisation du HTML, la fonctionnalité des différentes parties d'une page, la navigation sur le Web, l'utilisation de la typographie et des couleurs, la perception d'une page par le visiteur. Le livre part donc du point zéro pour conduire le lecteur, à travers un processus et des méthodes, jusqu'à l'achèvement d'un site. Les bases du code des feuilles de style en cascade sont simples. Mais pour les maîtriser, il faut en comprendre les objectifs et en décoder la grammaire.

Ce livre est construit, non pas comme une encyclopédie, mais comme un tutoriel. C'est non seulement un recueil de techniques et de pratiques, mais aussi un ouvrage de réflexions autour de la notion de design.

Comme nous le disions, le lecteur est censé connaître le HTML. Un débutant en CSS pourra comprendre notre propos grâce à la progressivité des connaissances distillées, tandis que le designer web déjà aguerri y trouvera matière à réflexions et exploitera avec profit exemples et astuces.

Au fil des pages, vous suivrez la construction d'un site, depuis le début de la conception, en passant par la mise en place du design et des feuilles de style. Vous apprendrez aussi à refondre un site avec les CSS sur le site photo-de-classe.com. Tout en suivant ce fil, le livre explorera les diverses possibilités et éclairera par des points théoriques les normes CSS.

Les CSS sont une tentative d'ordonner la jungle qu'est devenue le Web avec les navigateurs et leur tolérance aux erreurs et aux excentricités. Elles sont liées aux standards du Web. Il est donc nécessaire de parler du XHTLM et de l'accessibilité aux applications. La séparation du sens de la forme aboutit à la normalisation de l'un et de l'autre. D'un côté, les CSS permettent aux designers web de modeler les pages web à leur guise pour en faire parfois des chefs-d'œuvre de graphisme. De l'autre, les pages HTML se structurent et les développeurs explorent les promesses du Web sémantique, l'accession de tous aux contenus et les normes qui simplifient la construction des pages et leur maintenance.

1

Introduction

Les feuilles de style CSS (*Cascading Style Sheets*) sont un outil indispensable pour créer un site web, le maintenir et le rafraîchir régulièrement. La mise en écran web tend à se rapprocher de la mise en page d'un imprimé. Si les CSS favorisent ce rapprochement, elles le font avec quelques différences notables :

Tab. 1.1 : Différence entre le document écrit et le document web

Imprimé	Internet
Support figé du papier	Support hétérogène des navigateurs
Grand choix de polices	Choix limité de polices
Forme figée	Forme évolutive
Mise en page précise	Mise en écran personnalisable
Parcours de lecture linéaire	Parcours de lecture erratique
Fusion présentation/contenu	Séparation présentation/contenu

Ces différences changent complètement la perception du document. La volonté de se rapprocher de l'imprimé papier est un fantasme du développeur, alors que le communicateur préfère privilégier le message. Le Web d'ailleurs a eu une vocation initiale de communication. Cette vocation a vite évolué, dans un sens plus "artistique". Il s'agit de créer des œuvres d'art, ou du moins de belles pages. Tout développeur de pages web met le curseur à mi-chemin entre une communication claire et une belle page. L'idéal est d'arriver à concilier les deux. Tout "extrémisme" d'un côté ou de l'autre est source de frustrations pour l'internaute.

Le défaut d'un développeur trop carré est de proposer un texte noir sur un fond blanc pour avoir le contenu le plus clair, et d'utiliser au maximum toutes les possibilités du HTML comme les différentes tailles de titre, les listes numérotées ou à puces, etc. Le HTML a été inventé au CERN par des scientifiques à l'esprit rationnel et structuré, et non par des artistes. Au moment de son invention, le HTML servait surtout à modeler du texte. Les liens hypertextes sur un mot ou une expression permettaient d'avoir un accès direct à un texte de référence. Ils ont peu à peu "perverti" la lecture linéaire des textes.

Le défaut de "l'artiste" est de détourner le langage pour créer un labyrinthe à base de liens hypertextes, d'images et d'animations sur un contenu parfois confus et creux, surprendre le visiteur, jouer avec ses repères, souvent bousculer son parcours balisé et ses habitudes de lecture et de navigation.

Les deux tendances sont adéquates pour certains projets, mais la plupart du temps, l'idéal se situe entre les deux.

▲ Figure 1.1 : *Le curseur*

Très vite la forme a pris le dessus sur le contenu, les internautes se contentant de surfer sur la Toile, zapper et exploiter l'interactivité.

1.1 Évolution du HTML

Internet est né avec le lien hypertexte et le HTML, langage simple et démocratique, a pris le pas sur les langages d'expert comme le SGML. Le HTML était approprié pour des pages simples et tout le monde s'est mis à faire des pages à la main sur Notepad ou SimpleText. Le HTML a pour fonction de décrire une page à un navigateur. Les webmasters se sont sentis limités par rapport aux possibilités étendues qu'ils avaient avec la PAO (*publication assistée par ordinateur*). Le HTML s'est alors enrichi de nouvelles propriétés et de quelques balises. Les développeurs web avaient aussi recours à des astuces, comme les tableaux, qui permettaient de positionner des éléments sur la page alors que aucune balise ne le faisait. Le HTML mêle des indications de contenu (`<title>`) et de présentation (`<b>`) dans le même document, ce qui rend la maintenance d'un site lente et compliquée, voire impossible parfois. Le développeur doit se retrouver parmi toutes ces balises qui se chevauchent, sans compter les langages de script qui s'y ajoutent (JavaScript, PHP…). Dans la guerre qui opposait Netscape et Microsoft, chacun a fait cavalier seul, prenant l'initiative de créer de nouvelles balises et propriétés. La solution pour les

développeurs de sites, afin de remédier à cela, a consisté à utiliser seulement le code commun pour que leurs pages puissent être vues sur tous les navigateurs, et donc à se limiter au niveau des possibilités de mise en page, ou bien à faire un code pour chaque navigateur en le détectant par un code JavaScript ou PHP. D'où la nécessité de se mettre d'accord sur un standard commun avec le XHTML et les CSS.

Le XHTML est un puissant outil d'identification de contenus et assiste le développeur souhaitant créer des documents structurés, modulables et hiérarchisés. Il complète parfaitement les CSS qui, elles, gèrent la présentation de ces contenus.

1.2 Le tandem CSS et XHTML

Le tandem CSS et XHTML réconcilie les deux tendances du Web et marie ainsi un beau graphisme avec un contenu rigoureux et bien structuré.

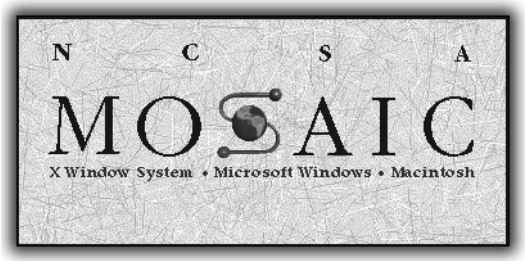
Les CSS ont été inventées par Håkon Wium Lie et Bert Bos en 1994-1995, puis ont été adoptées par le consortium W3C, qui s'est accordé sur un certain nombre de règles qui sont devenues la norme CSS1. Elles interviennent pour assouplir et alléger la création et la maintenance des sites web. Elles incarnent cette vieille idée de séparer le contenu et la présentation. Il est ainsi possible de modifier rapidement l'apparence d'un site en intervenant sur un seul fichier ou de disposer de plusieurs présentations pour un même fichier, selon le média ou le public visé. Ainsi les CSS sont associées à la notion d'accessibilité pour une présentation adaptée à chaque circonstance à partir d'un même contenu.

En allégeant les documents, le développeur a éprouvé le besoin d'avoir une page de contenu aussi bien structurée et aussi normalisée que la feuille de style. Il s'est alors tourné vers le XHTML. L'un des problèmes principaux auquel tout développeur de pages web est confronté est la diversité des navigateurs et de leur comportement dans l'"ingestion" et la restitution d'un même code.

1.3 Les navigateurs

En fait, Internet, tel que nous le connaissons, a vraiment commencé avec Mosaïc, le premier navigateur à gérer les formulaires et les images.

À l'époque, en 1993, il n'existait que deux cents sites et le code était simple. Puis sont arrivés Netscape et les autres navigateurs avec de nouvelles balises.



▲ Figure 1.2 : *Mosaïc*

Internet s'est développé grâce à la simplicité du code HTML et tout le monde s'est mis à créer des pages. Les navigateurs sont alors devenus tolérants aux erreurs de code sur les pages visitées. Par ailleurs, au fur et à mesure des versions des navigateurs, le code s'est complexifié, chaque navigateur se différenciant des autres par du code incompatible pour mettre leurs concurrents hors-jeu. Si la tendance s'était poursuivie, le Web serait devenue une sorte de Tour de Babel et aurait perdu son statut de réseau ouvert à tous. D'où le resserrement actuel autour des standards avec le XHTML.

```
<body bgcolor="#FFFFFF" background="backcours.gif"
text="#000000" link="#009900" vlink="#993300">
```

▲ Figure 1.3 : *La balise body*

La prise en compte des feuilles de style par les navigateurs a grandement allégé les pages de toutes les balises et propriétés qui décrivent la mise en page telles que `<font>`, `bgcolor`, etc. Les pages se chargent plus vite parce que le fichier HTML a maigri et que le fichier CSS est chargé une seule fois puis gardé en cache sur l'ordinateur de l'internaute.

1.4 Le concept de feuille de style

Le concept de feuille de style est une vieille idée. La plupart des logiciels de bureautique et de PAO utilisent ce principe car dès qu'un document atteint une certaine longueur, les styles se répètent sur toutes les pages. Il est donc plus intéressant de choisir les caractéristiques de chaque style une bonne fois pour toutes à partir du moment où chaque partie du texte est identifiée par sa fonction (titres, notes de bas de page, illustrations…). L'intérêt réside dans le fait que le texte est mis en forme automatiquement et qu'une modification sur cette feuille de style modifie tout le document immédiatement. L'évolution d'Internet appelait nécessairement ce genre de fonction. Cette fonction s'avère encore plus intéressante sur un site de plusieurs pages.

Les feuilles de style en cascade tirent leur nom de la notion d'héritage propre à ce type de feuille de style. Cette souplesse et cette simplicité d'usage ont permis à la première version des CSS de se répandre rapidement sur les sites web.

Les feuilles de style régissent les polices de caractères et leurs caractéristiques, la position d'un bloc, sa marge et le placement du contenu à l'intérieur, sa couleur de fond, la modification de l'aspect d'un lien au passage de la souris, les éléments de formulaire, la couleur de l'ascenseur de la page, etc.

Elles favorisent la fluidité dans l'espace, avec la possibilité d'avoir une feuille de style par média (écran, papier, mobile…), et bientôt par utilisateur, et aussi dans le temps, avec les modifications simplifiées lors de la mise à jour d'un site.

1.5 De la feuille de style à la feuille de style en cascade

Une feuille de style est un ensemble de codes qui identifie des balises grâce à des sélecteurs et des propriétés. Ce code est inclus dans l'en-tête d'une page HTML ou dans un fichier indépendant auquel peuvent se référer plusieurs pages web différentes, ou encore à l'intérieur d'une balise (inline ou en ligne). Ces trois modalités représentent des portées différentes pour le code.

En ligne

Le code inclus dans une balise comme propriété sera appliqué uniquement sur le contenu de la balise. Ce code est circonstanciel et ponctuel. Il est préférable de le mettre dans un fichier CSS ou dans l'en-tête du fichier, car ce type d'inclusion est difficile à modifier, surtout sur un gros site, parmi tous les fichiers et dans le code parfois très fourni de ces fichiers.

```
<div style="position: absolute; left: 10px; top: 100px;">
```

▲ Figure 1.4 : *Code en ligne dans une balise*

En-tête HTML

Le code inséré dans un en-tête HTML de fichier agit sur le contenu de ce fichier, et uniquement de ce fichier. Ce type d'inclusion est utilisé pour une présentation spéciale et circonstancielle qui se cantonne au fichier en question. Si ce n'est pas le cas, il est préférable de créer un fichier CSS indépendant.

```
<head>
/*(quelques lignes de code)*/
<style>
<!--
.menuskin{
  position:absolute;
  width:165px;
  background-color:#6699cc;
  border:2px solid black;
  font:normal 12px Verdana;
  line-height:18px;
  z-index:100;
  visibility:hidden;
}
.menuskin a{
  text-decoration:none;
  color:white;
  padding-left:10px;
  padding-right:10px;
}
#mouseoverstyle{
  background-color:#31185A;
}
#mouseoverstyle a{
  color:yellow;
```

```
}
-->
</style>
<head>
```

Fichier CSS

Le code entré dans un fichier indépendant peut servir plusieurs fichiers, voire un site entier, pour présenter leur contenu. C'est la solution optimale de séparation du contenu et de sa présentation. En une seule manipulation du fichier CSS, le webmaster modifie la présentation d'un grand nombre de fichiers qui en dépendent. Le fichier CSS peut importer plusieurs autres fichiers, portant l'extension *.css*, qui vont intégrer alors leurs sélecteurs à l'intérieur.

1.6 La notion de mise en page

Les règles de mise en page qui s'exercent dans l'impression d'un livre visent à améliorer la lisibilité du contenu. Il en est de même pour un site web, même si l'écran d'un ordinateur permet une plus grande liberté. Par ailleurs, le format d'un livre est la plupart du temps à la française ("portrait"), alors que celui d'un écran est à l'italienne ("paysage"), ce qui change le parcours de l'œil du lecteur.

▲ Figure 1.5 : *Le format à l'italienne*

Une page donne d'abord une impression d'ensemble créée par la disposition du texte, du blanc et des illustrations.

▲ Figure 1.6 : *Le format à la française*

Les maquettistes PAO ont l'avantage de connaître la surface précise de leur médium. Ils apposent donc sur leur page une grille virtuelle nommée "gabarit", qui leur permet de disposer les éléments d'une manière équilibrée (titrailles, paragraphes, colonnes, illustrations, notes, foliotage…). En revanche, l'écran d'ordinateur peut avoir des résolutions très différentes qui modifient la façon dont les éléments sont disposés sur l'écran. Le designer web doit surveiller trois points : l'harmonie des couleurs, la bonne répartition des blocs et la lisibilité du contenu.

Le lecteur lit une page dans le contexte du livre ou du journal, dans une continuité cohérente qui lui donne du sens, tandis que l'internaute peut entrer dans un site à partir de n'importe quelle page sans en appréhender le contexte, d'où la nécessité de répéter les éléments de navigation sur chaque page. L'internaute parcourt les pages au fil de sa pérégrination. Au départ, le seul navigateur s'appelait Mosaïc car le créateur des liens hypertextes avait rêvé d'avoir tous les documents ouverts en regard les uns des autres, alors que, dans la navigation Internet, le plus souvent, un document ouvert fait oublier le précédent. Le lien hypertexte n'est rien d'autre qu'une référence à un texte ou à une image, ou à tout autre document.

Le lien hypertexte et son usage

La limite de l'usage du lien hypertexte est son hypertrophie, qui fait ressembler parfois le Web à un labyrinthe. Une page qui contiendrait trop de liens hypertextes ressemblerait à un dédale où l'internaute se perdrait. À moins d'avoir une page de liens, il est conseillé de ne pas mettre une trop grande quantité de liens. En effet, si un internaute lit un article sur Internet qui renvoie des références trop nombreuses, il en oublie le texte de base, censé servir de colonne vertébrale à sa quête d'informations, et finit la lecture de la page avec une connaissance éclatée. Il vaut mieux ne pas surcharger le propos de liens. Insérer deux, trois liens dans le texte et un peu plus à la fin ou sur le côté suffit pour éclairer le propos d'une page. Par contre, le visiteur s'attend à disposer de nombreux liens sur la page d'accueil.

▲ Figure 1.7 : *L'intitulé de lien*

Il existe un art du lien, qui donne envie de cliquer, un art d'introduire le lien par un contexte, un classement ou par un intitulé explicite.

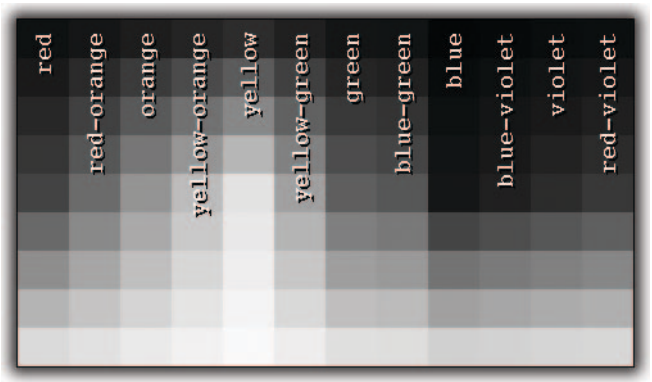
Les liens peuvent s'ouvrir sur une nouvelle fenêtre. L'abus de cette pratique rend la communication confuse et peut, en outre, agacer les internautes qui se retrouvent avec un grand nombre de fenêtres ouvertes et non reliées entre elles. Par ailleurs, de nombreux internautes peuvent cliquer du bouton droit sur un lien pour ouvrir une nouvelle fenêtre quand ils en ont envie.

Précision du lien

Le lien doit renvoyer sur la référence précise, et non pas sur une page où le lecteur doit chercher la référence dans un ensemble d'informations.

L'harmonie des couleurs

Les couleurs donnent la première impression au visiteur, au moment de l'ouverture de la page. Ensuite, elles doivent être suffisamment discrètes pour accompagner la lecture, conduire l'œil sans interférer. La couleur a le même rôle que la musique de film : elle souligne les espaces importants, elle tonifie ou s'efface, mais surtout elle sait se faire oublier. Pour être sûr d'afficher des couleurs fidèles sur tout écran, il est conseillé de se servir d'une palette de couleurs sécurisée. Les couleurs web sécurisées regroupent les 216 couleurs utilisées par les navigateurs, quelle que soit la plate-forme employée.



▲ Figure 1.8 : *La palette des couleurs*

Cette palette sécurisée garantit que les couleurs s'affichent correctement avec la définition de l'écran réglée sur une profondeur de 256 couleurs. Pour créer une image légère, il faut se contenter des 256 couleurs. Quand un écran ne peut afficher la couleur qui lui est demandée, il affiche alors une couleur voisine.

Tab. 1.2 : Affichage des couleurs à l'écran

Bits par pixel	Nombre maximum de couleurs
1	2
4	16
8	256

Tab. 1.2 : Affichage des couleurs à l'écran

Bits par pixel	Nombre maximum de couleurs
16	65 536
24	16 777 216

La plupart des écrans à l'heure actuelle sont de 24 bits et donc susceptibles d'afficher la couleur demandée.

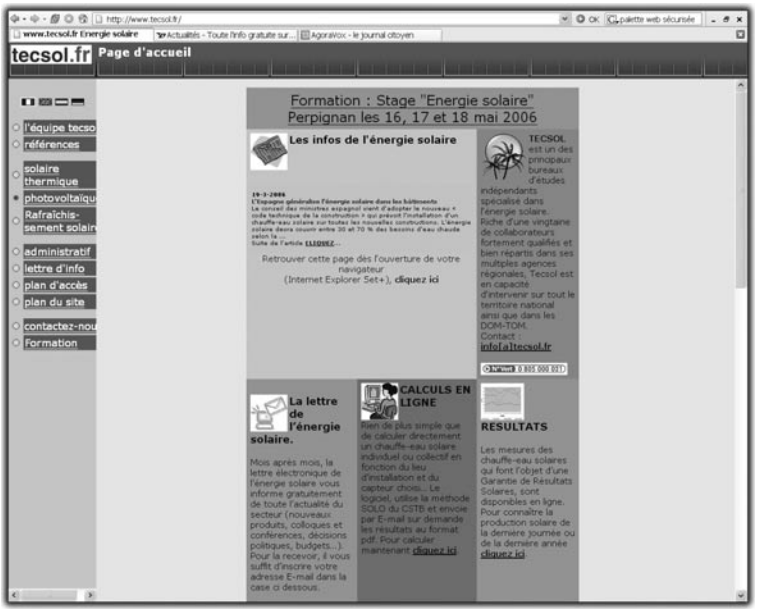
Tab. 1.3 : La mémoire graphique et la résolution d'écran

Résolution et mémoire	512 Ko	1 Mo	2 Mo	4 Mo	8 Mo
640 × 480	8 bits	24 bits	24 bits	24 bits	24 bits
800 × 600	8 bits	16 bits	24 bits	24 bits	24 bits
1 024 × 768		8 bits	16 bits	24 bits	24 bits
1 280 × 1 024			8 bits	24 bits	24 bits
1 600 × 1 200			8 bits	16 bits	24 bits

Selon le thème du site, une couleur ou deux seront privilégiées. Par exemple, pour le thème du nautisme, il pourrait s'agir du bleu et du jaune ; pour la nature, le vert est indiqué. Certaines couleurs sont froides, comme le bleu, et d'autres chaudes, comme le jaune et le rouge, ou encore fraîches, comme le vert. Il existe par ailleurs de nombreuses nuances. La couleur affectée au site comme dominante lui donnera une température et influencera le visiteur, sans même qu'il ne s'en aperçoive.

La répartition des blocs

La répartition des blocs s'inscrit sur le fond de page. Ils sont matérialisés par des fonds de couleur, des filets, des gouttières, et leur ensemble forme un motif qui aide le visiteur à promener son regard, à s'arrêter ici ou là.



▲ Figure 1.9 : *Les blocs*

Si le fond de page est suffisamment clair et la proportion entre le contenu et le "blanc" équilibrée, le lecteur a plaisir à revenir sur cette page. Les proportions traditionnelles entre le texte et le blanc vont de 5:8 pour de l'artistique jusqu'à 3:4 pour du texte fourni. Le standard est entre les deux, 2:3.

Le blanc est représenté par les espaces (gouttières, interlignage, fond de page).

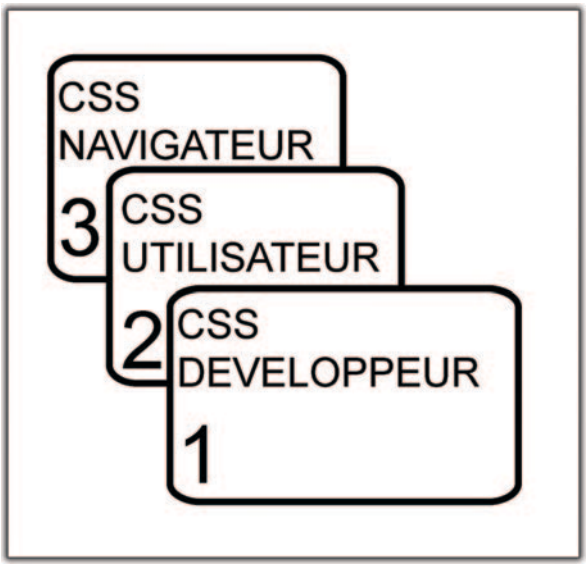
La lisibilité du texte

Le texte sur un écran est moins lisible que sur le papier, l'écran étant constitué d'un ensemble de carrés appelés "pixels". Le corps des polices employées ne doit pas être trop petit, surtout pour les polices avec empattement. Cette règle se renforce avec une police claire sur un fond sombre, qui demande souvent d'être en gras.

1.7 Les cascades de style

Internet a emprunté à l'imprimerie un certain nombre de métaphores. On parle de pages web et de mise en page web. Les designers web viennent souvent du monde de l'imprimerie, et pourtant, les techniques de PAO employées dans l'imprimerie ne peuvent s'appliquer entièrement sur la Toile. Le Web est un monde mouvant, la page web se plie à de nombreux médias. La même page vue par des navigateurs différents aura une apparence modifiée, autant par les polices de caractères que par les couleurs et par la mise en page elle-même. La mise en page est réalisée par trois acteurs :

- le développeur du site, qui peut avoir créé une ou plusieurs feuilles de style pour la même page ;
- l'utilisateur qui peut avoir conçu une feuille de style utilisateur pour normaliser les pages qu'il visualise et les rendre plus claires à lire par exemple, ou encore pour leur donner un design qui lui plaît.
- le navigateur qui utilise une feuille de style par défaut.



▲ Figure 1.10 : *La priorité des différentes feuilles de style*

Le résultat est la combinaison de ces différentes feuilles de style, bâtie dans la mémoire du navigateur et créée au fur et à mesure de la lecture de toutes les feuilles de style.

Chacun de ses acteurs propose sa propre feuille de style pour une page donnée. Ainsi, déjà avec le CSS 1, le navigateur corrigeait le code envoyé par une feuille de style et en rectifiait les défauts susceptibles de rendre la page illisible, comme des lignes qui se chevauchent à cause d'une hauteur de ligne inadéquate par exemple.

Avec la version 2 des feuilles CSS, l'utilisateur peut avoir sa propre feuille de style, qui s'ajoute à celle créée par le développeur et à celle du navigateur. Ces trois feuilles peuvent contenir des informations contradictoires, la feuille de style développeur indiquant par exemple que les liens sont verts, et celle de l'utilisateur que les liens sont rouges. Ces indications vont déclencher une contradiction qui sera gérée par le navigateur dans le sens du développeur. Ainsi les indications sur la feuille de style développeur prendront le pas sur la feuille de style utilisateur.

Quelle est la feuille de style la plus "écoutée" par le navigateur ?

En cas de conflit de sélecteur ou de propriété, la feuille proposée par le développeur du site prendra le pas sur la feuille de l'utilisateur, qui prendra le pas sur la feuille de style par défaut du navigateur. Cet ordre sera modifié si le mot-clé `!important` est utilisé.

L'utilisateur a la possibilité d'ajouter le mot-clé `!important` pour s'assurer que le navigateur tiendra compte en priorité de certains éléments.

```
body { background-color: white !important; color: black !important}
```

Le code précédent affichera des pages sévères, mais très lisibles pour quelqu'un qui confond les couleurs, comme un daltonien, ou quelqu'un qui a une mauvaise vue. Même si le développeur a écrit un beau code qui envoie une belle page au navigateur, quand bien même ce développeur aurait inscrit `!important` sur chacune de ses lignes dans le fichier CSS, les indications de l'utilisateur supplanteront toute sa belle construction.

Avec l'arrivée des nouveaux navigateurs compatibles CSS2, le rapport au Web change pour l'utilisateur. Ainsi, quand un site ne lui plaît pas, il peut le modifier. Le développeur doit en tenir compte et éviter une mise à l'écran trop rigide.

Face à cela, le développeur a deux options :

- Soit il verrouille sa mise en page car il attache plus d'importance au graphisme.
- Soit il fait une mise en page ouverte, qui s'adapte à chaque page, car il attache plus d'importance au contenu.

Dans ce qui se profile, la page web sera de moins en moins celle du développeur, et de plus en plus celle du lecteur. Libre à lui de refuser ou d'accepter le style graphique du développeur. Mais si la page obéit aux normes XHTML, elle restera lisible, quels que soient les aspects.

1.8 Le XHTML

Le XHTML est une normalisation du HTML, qui le rapproche du XML. L'application du code est plus stricte. Un document XHTML du fait de la rigueur de sa grammaire permet de faire analyser son code par des validateurs qui en relèvent les erreurs.

Le XHTML est adéquat pour les CSS dans le sens qu'il donne des directives au navigateur pour bien appliquer les CSS. Il simplifie le travail des navigateurs et celui des robots, qui écument le Web pour les moteurs de recherche. Une des caractéristiques veut que l'on peut inventer des propriétés pour marquer le contenu avec son identifiant.

Le XHTML met en évidence la structure en normalisant et en dépouillant le code. Pour écrire un article, il faut fournir un contenu à chaque marqueur (titre 1 par exemple), ce qui oblige le rédacteur à construire son article et à le penser pour le Web. Un article imprimé est difficilement transposable directement sur une page web ; il demande quelques transformations.

1.9 La structure d'une page

L'usage de l'imprimé a forgé, au fil du temps, des règles communes propres à chaque langue. Ces règles concernent la typographie et la mise en page, mais aussi le contenu d'une page.

Écrire pour le Web

On n'écrit pas pour le Web comme on écrit pour un livre, un journal ou un magazine. La lecture à l'écran est plus malaisée que la lecture d'un imprimé. Aussi est-il nécessaire de faciliter la lecture en structurant l'article avec des titres, sous-titres, intertitres, résumés, accroches. Il est préférable de ne pas utiliser plus de trois polices de caractères et d'emprunter le même modèle pour chaque page car le lecteur se familiarisera ainsi avec la manière de procéder sur le site.

Pour lire un journal, le lecteur parcourt d'abord la page puis se concentre sur le titre qui l'a accroché. Ensuite, dès le chapeau, puis le premier paragraphe, il décide s'il lit attentivement l'article ou s'il le survole simplement pour en tirer l'essentiel.

▲ Figure 1.11 : *Les blocs*

L'article pour Internet doit subir une réduction, avec des phrases courtes qui s'enchaînent logiquement. Une idée par phrase et un thème par paragraphe sont deux directives à respecter.

Certaines phrases pourront être mises en forme de listes ou de schémas. Parfois, pour simplifier un texte, il suffit de le décliner sous plusieurs formes qui se compléteront. Ainsi, chaque page sera courte et sa mise en forme plus attractive. Des encadrés, des intertitres, une FAQ, une page par thème aideront l'internaute à s'orienter dans un sujet, plutôt qu'une page interminable. Une expression en italique, en gras, un point d'exclamation serviront de repère au lecteur. Il ne faut pas hésiter à se répéter sous diverses formes.

La lecture sur le Web est avant tout informative, dans le sens où l'internaute parcourt des yeux un texte pour y trouver les informations qu'il cherche. La lecture ne se fait pas mot à mot parce que le Web n'est pas adapté à ce type de pratique, même si cela est possible. L'écran est plus adapté aux images qu'au texte. Le caractère de l'écran n'est qu'une imitation du caractère d'imprimerie, une émulation du livre papier. L'écran est de 96 dpi (points par pouce) alors que l'impression du livre est au minimum de 300 dpi. Une œuvre narrative, comme un roman, n'est pas adaptée à l'écran et, au bout de quelques pages, l'impétrant sera poussé à imprimer le texte pour le lire plus aisément, sans s'arracher les yeux.

La lecture sur le Web est active, grâce aux moteurs de recherche dans le texte, aux incursions vers d'autres textes via les liens de la page et à la communication avec l'auteur par l'intermédiaire d'un courriel.

Imprimer et relire

Rien ne donne plus mauvaise impression que des fautes d'orthographe dans un beau site. Il est recommandé d'imprimer les pages web pour les relire avec attention.

L'information sur le Web est parcellaire et éphémère. Elle doit souvent être morcelée en plusieurs pages qui communiquent par les liens de navigation. C'est un écho au phénomène du zapping, avec la télécommande de la télévision. Le consommateur glane à droite et à gauche les informations qu'il désire. L'information est éphémère, ce qui donne du

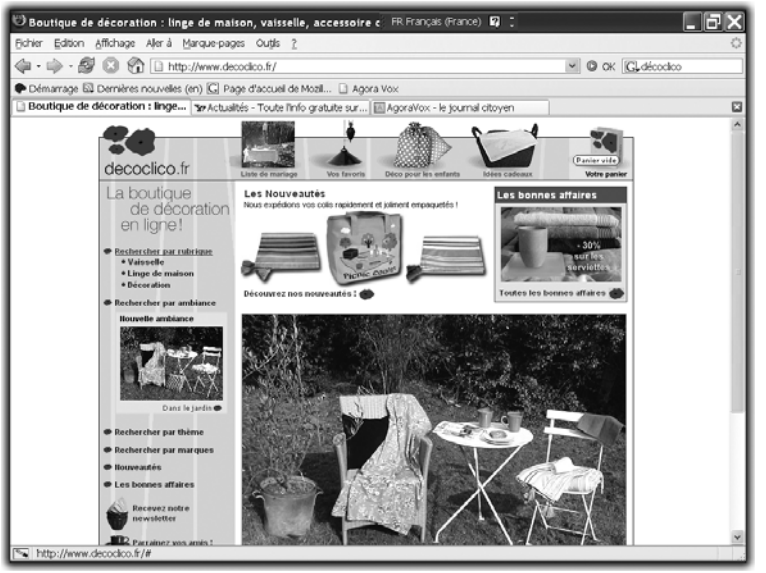
Web, l'impression d'un chantier permanent. Ce n'est pas simplement ces panneaux "En travaux" ou autres sur les sites qui confortent cette impression, mais cette facilité de mise à jour, alors qu'un livre ne peut être mis à jour qu'au gré de ses rééditions. Une adresse de site ou de page peut devenir vide et inexistante tandis que des moteurs de recherche comme Google gardent l'archive de l'ancienne page, qui ressuscite au hasard d'une recherche par mots-clés.

L'information sur le Web est imprévisible. Internet emploie de nombreuses métaphores pour décrire l'expérience de la connexion sur le Web : navigation, navigateur, surf, des termes qui dépeignent un voyage sans but, une promenade au gré des liens et des idées qui traversent l'internaute et qu'il saisit dans le formulaire Google inséré dans son navigateur.

Hiérarchisation de l'information

Selon les études menées pour espionner le regard de l'internaute, il s'avère que la partie de l'écran vue en premier se situe à gauche. Par conséquent, l'information la plus importante se situe en haut de page, principalement à gauche, aux trois quarts de l'écran, et occupe une place conséquente, tandis que l'information la moins importante est située en bas à droite. En se rendant sur un site de news, comme Google News ou Yahoo!, il est facile de vérifier où se situe la nouvelle qui saute aux yeux (voir fig. 1.2).

Le choix judicieux des titres et intertitres contribue à la bonne compréhension de la page en jalonnant le texte de repères. L'utilisation de couleurs pour les titres ou en arrière-plan pour en distinguer les blocs améliore le parcours du lecteur. Tout ce qui est facile, immédiat et reconnaissable favorise la compréhension et l'approche d'un site.



▲ Figure 1.12 : *La hiérarchie dans l'information*

Communication plus simple entre les machines

Un langage normé peut sembler plus rigide. En fait, il a plus de souplesse. Il est plus facile d'agréger plusieurs sources d'informations en syndication. Par ailleurs, un tel langage se plie à tous les médias. Quand une grammaire est respectée, la mise en forme est plus prévisible et donc le contenu est plus facile à utiliser par des machines et des logiciels. Étant donné le volume d'informations exponentiel véhiculé par Internet, il est naturel de structurer l'expression pour en tirer le maximum de sens et aider les moteurs de recherche, qui sont l'outil indispensable à l'internaute pour trouver une information pointue dans cette masse de données. Le XHTML est basé sur le XML, qui est en quelque sorte l'espéranto informatique, un langage passerelle qui permet, dans un environnement hétérogène, aux applications de communiquer entre elles. Le XHTML est prévisible et il est donc plus facile de systématiser des opérations sur les données pour en tirer la substantifique moelle. C'est sans doute dans cette direction qu'ira une partie du développement de l'Internet de demain.

1.10 Conclusion

Le XHTML et les CSS promeuvent une autre pratique de l'écriture et du graphisme au service de la communication avec les humains, à travers l'ergonomie et l'"utilisabilité", et entre les machines, par le moyen d'un code propre et d'un contenu bien structuré qui "parle" à tous les navigateurs actuels. L'agencement de la couleur, la position des textes et des images, la distribution équilibrée des liens hypertextes, un contenu écrit pour l'écran amènent clairement plus d'internautes sur certains sites, qui ont une vision à long terme et gagnent en visibilité sur la Toile.

2

La réflexion

Le site web est un outil qui sert à communiquer. Avant de le réaliser, il est donc essentiel de connaître le public visé car ce dernier donnera l'orientation de la charte graphique et le ton du contenu. Si vous vous adressez à une tranche 15-24 ans avec la musique rock comme thème, vous aurez un ton familier, avec le vocabulaire adéquat, pour créer un lien immédiat avec ce public. Le design d'un site parle immédiatement au visiteur, mais il est le résultat d'un travail en amont.

Le site est donc au cœur d'un projet et le projet ne se construit pas au fur et à mesure de l'élaboration du site. Ce projet doit se voir au premier coup d'œil, sur la page d'accueil. Internet relie des millions de sites ; il est donc nécessaire de se démarquer des autres sites par une intention originale et annoncer immédiatement le contenu et le ton.

2.1 Les types de sites

Il existe plusieurs types de sites selon l'objectif visé. Ils demandent une charte graphique différente. Par exemple, le site vitrine est avant tout un travail de graphiste. Le contenu doit être court et efficace, alors que le portail communautaire sera davantage centré sur l'interactivité.

Les sites vitrine ou sites plaquette

Les sites vitrine ou sites plaquette ont comme objectif de présenter une société sur Internet pour des internautes qui ne connaissent pas la société. Ils contiennent quelques pages présentant les produits et services, la situation géographique, une page contact, des points pratiques (heures d'ouverture, plan d'accès…). Cette catégorie de sites reprend souvent le graphisme de la plaquette de la société et leur interactivité est restreinte à la page contact avec éventuellement un formulaire.

Les internautes viennent y chercher une information précise s'ils connaissent l'entreprise. Ils peuvent découvrir le site à travers un moteur de recherche parce qu'ils recherchent un produit ou un service local. Le site n'est pas destiné à apporter toutes les informations mais à donner envie au visiteur d'en savoir plus en téléphonant ou en rendant visite à l'entreprise. Le design doit être agréable, clair, avec des polices faciles à lire, une

navigation classique et un accès simple aux pages. `www.nexusdev.com` constitue un exemple de site vitrine bien conçu et complet. Il est mis à jour fréquemment et constitue une bonne vitrine pour la société Nexus Développement.

▲ Figure 2.1 : *Le site de Nexus Développement*

Les sites catalogue

Le premier objectif d'un site catalogue est d'éviter, trivialement, l'envoi d'un catalogue par la poste. L'autre objectif est que le chaland voit immédiatement les produits pour un achat d'impulsion. L'article ne sera pas acheté sur le site, mais le visiteur pourra se faire une idée et se déplacer. Tous les échanges s'accélèrent et le consommateur est sensible à la frustration, qui a comme effet qu'il ira voir la concurrence plutôt que d'attendre.

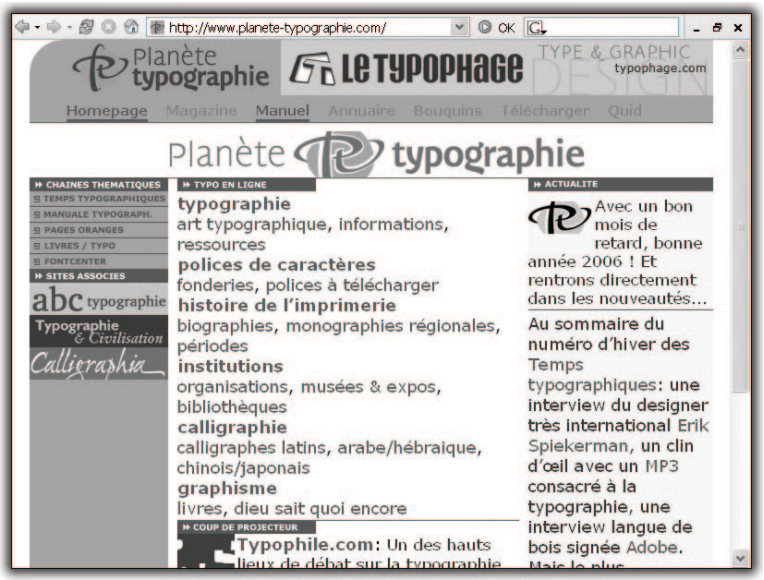
Ces sites épousent la mise en page du catalogue de produits du magasin. Parfois, comme sur `www.feuvert.fr`, les pages sont feuilletées comme un catalogue papier.

▲ Figure 2.2 : *Le site de Feu Vert*

Le site catalogue ne permet pas d'acheter un produit en ligne. Il doit être ergonomique pour que le produit soit facile à trouver et ne nécessite pas de feuilleter tout le catalogue. Les pages ont parfois tendance à être lourdes, ce qui a pour effet d'agacer le visiteur. Il est judicieux, en plus de classer les produits en de nombreuses catégories, d'insérer un moteur de recherche avec plusieurs modes de recherche (mots-clés, catégorie, référence produit, nom du produit…). La documentation des articles peut être présente sur une page web ou un document à télécharger. Une foire aux questions (FAQ), avec un formulaire pour le SAV, pourront être proposés.

Les sites d'information

Les sites d'information approfondissent un thème et diffusant des informations avec une visée professionnelle. Les pages doivent être dépouillées pour en faciliter la lecture, avec un beau graphisme restant sobre. Le design est là pour souligner le contenu mais c'est le contenu qui importe le plus. Les liens de navigation sont explicites. Un exemple de beau site d'information sur la typographie est `www.planete-typographie.com`.



▲ Figure 2.3 : *Le site Planète Typographie*

Les sites marchands

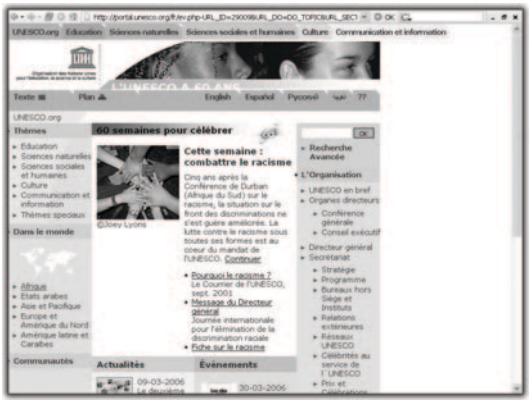
Les sites marchands vendent directement leurs produits sur Internet avec la possibilité de payer en ligne ou par un autre moyen. Le design dépend du type de produit vendu et de la ligne de marketing suivie. Un site qui construit toute sa publicité sur les prix va mettre en avant les promotions sur sa page d'accueil et tenir au courant les inscrits par un mailing

régulier, comme le fait `www.bexley.fr`. Un site comme `www.outilssolaires.com`, qui présente des produits dans une niche et sans vraie concurrence, mettra plutôt les nouveautés en avant et, parfois, se rapprochera d'un portail communautaire dans la mesure où cette communication entre les membres génère du chiffre d'affaires ou tout simplement par passion.

▲ Figure 2.4 : *Le site Outils Solaires*

Les sites institutionnels

Les sites institutionnels présentent l'organisation et leurs valeurs fondatrices. Ils ont un souci de clarté. La page d'accueil affiche généralement les dernières nouvelles, avec une colonne de liens sur la gauche qui mènent aux ressources sur l'organisation interne d'une part, et aux ressources diverses sur leur principal thème d'autre part. Le site de l'Unesco (`http://portal.unesco.org/fr`) en est un exemple.



▲ Figure 2.5 : *Le site de l'Unesco*

Les sites personnels

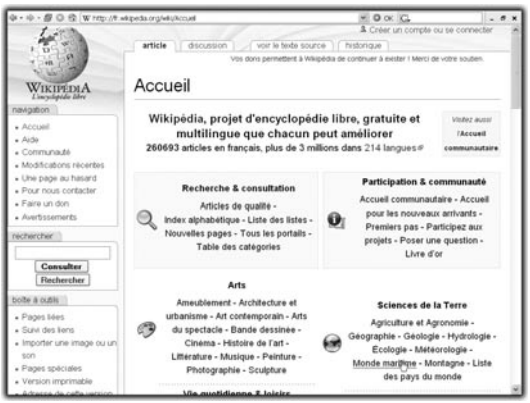
Ces sites sont réalisés par des amateurs, poussés par leur passion pour un sujet. Ils sont multiformes, divers et changeants. C'est le terrain d'expérimentation des développeurs en herbe. Certaines de ces adresses, au fil des métamorphoses, deviennent des sites d'information ou des sites communautaires. Par définition, ils n'ont pas de forme définie et permettent la plus grande originalité, à l'image de `http://lesmains.free.fr`.

▲ Figure 2.6 : *Le site Les Mains*

Les sites communautaires

▲ Figure 2.7 : *Le site Agoravox*

Les sites communautaires sont des portails qui permettent à des internautes de partager un intérêt commun, avec beaucoup d'interactivité. Ils sont faits par les internautes eux-mêmes, le webmaster n'étant là que pour animer et lancer une dynamique. Leur ambition est de réunir le maximum de passionnés et de devenir incontournables sur le sujet traité, comme Wikipédia `http://fr.wikipedia.org/wiki/Accueil` ou encore `www.agoravox.fr`.



▲ Figure 2.8 : *Le site Wikipédia*

Les sites intranet et les sites extranet

Les sites intranet sont accessibles de l'intérieur d'une entreprise ou d'une direction. Ils ont pour objet la mise à disposition et le partage d'informations professionnelles. Leurs pendants sont les sites extranet sur le Web qui, grâce à une authentification, permettent à des salariés d'une entreprise d'accéder à une interface personnalisée et d'administrer des informations. Ils peuvent cibler les fournisseurs, les acheteurs, les commerciaux itinérants, etc.

2.2 Quelques règles générales

Le site doit être centré sur l'utilisateur visé. Ses fonctions vitales (navigation, contenu…) doivent être visibles pour toutes les configurations. L'internaute a peu de patience et vous le perdrez s'il fait face aux situations suivantes :

- Il doit chercher où cliquer.
- Il ne voit pas dès la première page le thème du site au-delà de son titre.
- Les caractères sont petits et il ne peut les agrandir avec son navigateur.
- Les couleurs parasitent une bonne lecture.
- Le site est conçu avec des couleurs fluorescentes et dissonantes.
- Une partie de l'écran n'est pas accessible.
- La page est trop lourde à charger ou, pire, fait planter le navigateur.
- La page demande une extension du navigateur que le visiteur ne possède pas.

Le concepteur du site se met à la place du visiteur lambda. L'ordinateur et la technique doivent se faire oublier pour que l'internaute se plonge dans le site et vive une vraie expérience qui lui donnera envie de revenir.

Nous allons prendre en exemple la réalisation d'un site pour un gîte rural.

2.3 Le site

Le but du site des Jardins d'Écoute S'il Pleut est de se faire connaître, de constituer un repère sur Internet pour les visiteurs qui le connaissent déjà et enfin d'exposer la philosophie de l'entreprise, qui développe trois pôles :

- les gîtes ;
- la pépinière ;
- le conseil.

La difficulté consiste à réunir ces trois activités sans qu'il n'y ait de hiatus entre elles. Le but est donc de trouver une charte homogène, qui différencie cependant les trois pôles. La gageure est ici de faire en sorte que chaque partie du site trouve son public, ou plutôt que chaque public identifie bien ce qu'il vient chercher sur le site.

▲ Figure 2.9 : *Écoute S'il Pleut*

2.4 Le public

À qui le site est-il destiné ? Il s'adresse aux curieux, aux personnes indécises quant à leurs prochaines vacances, qui cherchent à passer un séjour convivial, au calme. Il doit contenir de nombreuses photos et mettre en valeur le gîte, l'accueil et les activités possibles dans les environs. Un

gîte rural n'est pas différent d'un hôtel ; il est simplement plus convivial et cultive davantage la proximité.

▲ Figure 2.10 : *Écoute S'il Pleut, gîtes*

Le public a besoin d'être séduit et rassuré, de connaître les propriétaires du gîte, la région, les activités. Nous allons faire ensemble le site les Jardins d'Écoute S'il Pleut (`www.ardaee.com`), un lieu qui conjugue horticulture, jardinage et accueil. L'accent sera mis sur le gîte.

Pour cibler les futurs visiteurs, il est intéressant de créer des profils types et de dresser la priorité de chacun. Plusieurs profils sont envisageables :

- personnes ou familles en recherche de gîtes accessibles aux handicapés, portant le label "Tourisme et handicap" et "Gîtes de France", dans le cadre d'une ferme horticole ;
- personnes ou familles en recherche d'un lieu d'accueil pour courts séjours de repos ;
- familles britanniques ;

- jardiniers, amateurs ou confirmés, intéressés par la production de plantes vivaces, botaniques et rustiques, selon des méthodes biologiques, et voulant découvrir et partager les passions des hôtes ;
- personnes en recherche d'aide ou d'informations sur le floraison de jardin, ou sur des plantes vivaces, botaniques, rustiques ;
- personnes intéressées par notre démarche de pluri-activités autour de la ferme horticole, de l'association ARDAEE et de ses services d'ingénierie ;
- personnes intéressées par l'environnement et le sud de la Vendée, avec forêts, lacs, monuments, lieux de pêche, vie culturelle.

Le site devra donc donner à ce public hétérogène les informations qu'il attend et faire en sorte qu'elles soient aisément accessibles.

2.5 La carte du site

Élaborer la carte du site consiste d'abord à inventorier toutes les pages puis à les classer en les hiérarchisant. Cette carte est une réflexion sur les objectifs du site et le contenu dont nous disposons. Ce travail s'élabore avec le client et permet d'estimer l'ampleur de la tâche.

Le site présente le lieu et sa région, les activités, les événements annuels, les coins remarquables dans les environs, les restaurants, etc.

Le principe est de commencer par la page la plus importante, celle que les internautes viennent visiter quand ils cliquent sur le lien qui mène au site ou quand ils saisissent l'adresse dans leur navigateur. Voici une photo avec une brève présentation du lieu (voir Figure 2.11).

Le visiteur parcourt les pages du regard. Il est important de hiérarchiser l'information pour donner une direction à la navigation, en mettant le lien le plus important bien en vue, soit au-dessus des autres, soit à un endroit où il sera repéré en premier.

Les liens répertoriés sont les suivants :

- *Les gîtes* : une page par gîte ;
- *Les pépinières* (méthodes de production, plantes, vente) ;
- *Conseil* ;

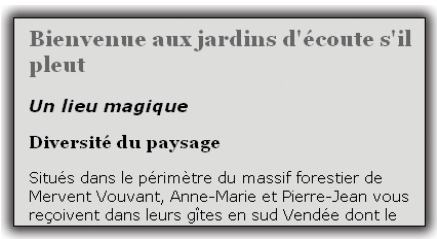
◄ Figure 2.11 : *Écoute S'il Pleut, liens*

- *Contact* ;
- *Partenaires* ;
- *Liens amis* ;
- *Comment venir ?*.

Par la suite, la carte du site pourra évoluer, mais en l'état, elle constitue une base pour facturer les services au client, le travail est facturer sur le temps évaluer pour réaliser cette arborescence ; tout ce qui s'ajoutera par la suite sera facturé en plus.

2.6 Identifier le contenu sémantique

Chaque page contient un titre principal et au moins un autre secondaire. Sous un titre, quand le texte de la page est long, il est souvent utile de mettre un résumé, introduisant les sujets évoqués dans ce qui suit. L'article est entrecoupé de sous-titres. La page contient aussi un ensemble de mots-clés, qui aideront le visiteur à identifier les thèmes et permettront au site de bien être référencé par les moteurs de recherche.



◄ Figure 2.12 : *Écoute S'il Pleut, contenu sémantique*

2.7 La navigation et la disposition des pages

La barre de navigation doit être visible et bien en évidence. Dans ce domaine, il vaut mieux rester classique car naviguer sur un site est un geste familier et ne doit pas devenir une quête du Graal. Certains sites sont tellement sophistiqués que les liens de navigation se fondent dans l'arrière-plan ou sont si peu conformes aux recommandations W3C que le navigateur confond les diverses boîtes `div`, qu'il superpose sans vergogne.

2.8 Choisir les outils

Un outil gratuit qui simplifie la création et la modification des CSS est TopStyle Lite. C'est la version gratuite de TopStyle Pro. Le logiciel est ergonomique et simple à utiliser.

Trois types d'outils sont indispensables :

- un outil pour les feuilles de style ;
- des outils pour la couleur ;
- un outil pour l'icône.

TopStyle Pro

Écrit par le créateur de HomeSite, TopStyle est un bon éditeur de CSS, qui manie bien le HTML et le XHTML. Il est téléchargeable sur `www.bradsoft.com/download`. Il existe des versions payantes et une gratuite, TopStyle Lite, qui possède une fonction de validation de style selon les différentes normes CSS (voir Figure 2.13).

TopStyle Pro propose, en plus, des outils de validation de HTML et de XHTML Une fonction permet aussi de normaliser un code en un clic, avec la fonction de vérificateur de mise à jour (Style Upgrade), en transposant les balises obsolètes (balises de police…) dans la feuille de style. La fonction de vérification de style inspecte le style CSS en fonction aussi de la compatibilité avec les différents navigateurs.

TopStyle accompagne en toute transparence les logiciels Dreamweaver MX et HomeSite.

▲ Figure 2.13 : *TopStyle*

Master Styler

Master Styler est un outil pratique pour manipuler le contenu des feuilles de style et identifier certains sélecteurs. Les erreurs sont mises en évidence par un surlignage. Le résultat est visualisé dans différents navigateurs, à condition que ces navigateurs soient installés. L'interface est ergonomique et agréable. Il est téléchargeable sur `www.westciv.com/style_master`, avec une version d'évaluation qui devient payante un mois après.

▲ Figure 2.14 : *Master Styler*

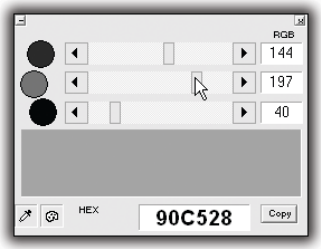
ColorZilla

ColorZilla est une extension de Firefox, qui fournit une pipette pour copier le code d'une couleur sur le Web et qui peut aussi mesurer la distance en pixels entre deux points. C'est une icône qui s'affiche dans la barre d'état, au bas de Firefox. ColorZilla est téléchargeable sur `https://addons.mozilla.org/extensions/moreinfo.php?id=271`

▲ Figure 2.15 : *ColorZilla*

Colourificator

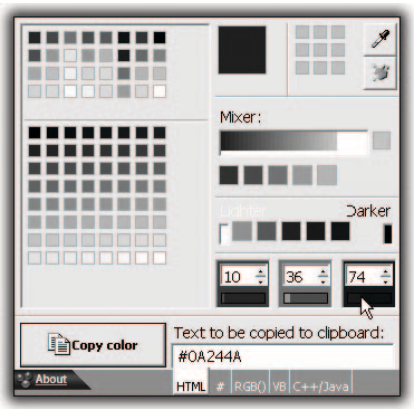
Colourificator est une petite palette pour PC, dotée d'une pipette qui permet de recueillir n'importe quelle couleur affichée à l'écran. Les couleurs sont codées en RVB et restituées en hexadécimal ou en pourcentage ou en valeurs de rouge, vert et bleu jusqu'à 255. La couleur peut être ensuite copiée dans le Presse-papiers via un bouton. Cette palette peut être conservée au-dessus des autres fenêtres, y compris de la fenêtre active. Ce petit logiciel est extrêmement pratique. Il est téléchargeable sur `http://colourificator.sitekit.net`.



◄ Figure 2.16 : *Colourificator*

EclipsePalette

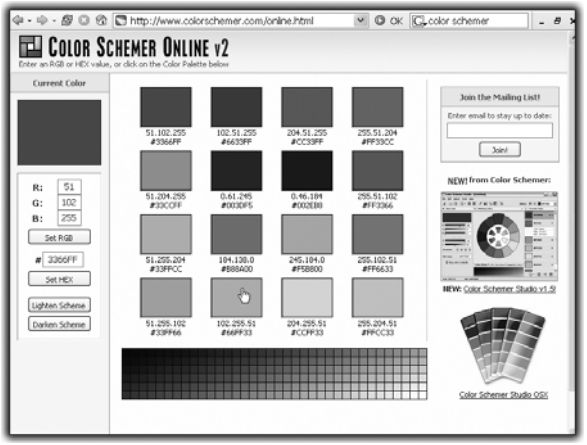
EclipsePalette est l'équivalent de Colourificator, avec une interface moins ergonomique et moins pratique, mais avec la possibilité d'éclaircir ou de foncer la couleur récoltée par la pipette. Un mélangeur permet de créer un dégradé entre deux couleurs. EclipsePalette est téléchargeable sur `www.greeneclipsesoftware.com/eclipsepalette.html`



◄ Figure 2.17 : *EclipsePalette*

Color Schemer

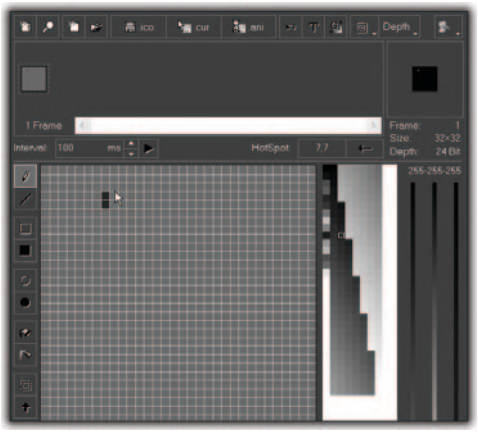
Color Schemer est un logiciel qui se sert des lois de l'harmonie des couleurs pour proposer à partir d'une couleur, toutes celles avec lesquelles elle se marie bien. Il est possible d'éclaircir ou assombrir une couleur pour créer un dégradé ou d'avoir un florilège de couleurs. Color Schemer est payant et téléchargeable sur `www.colorschemer.com/download.php`.



▲ Figure 2.18 : *Color Schemer*

IconArt

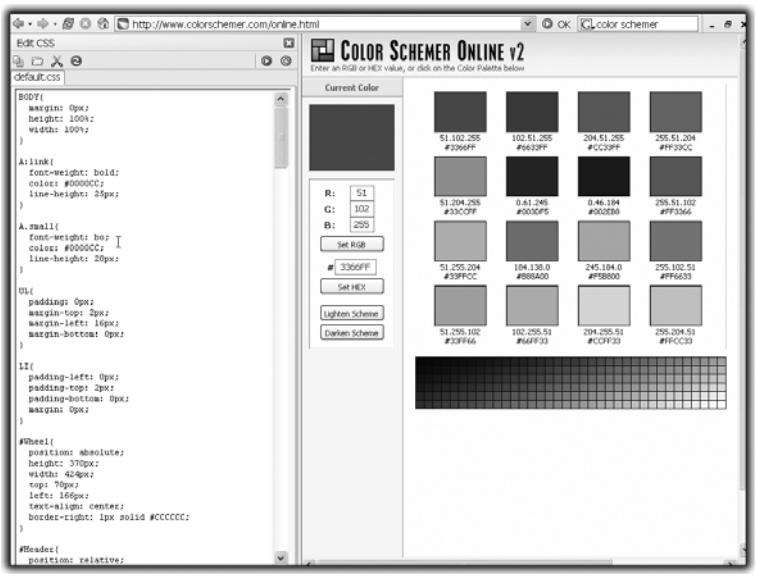
Pour avoir une identité graphique dans la barre d'adresse d'un navigateur ou dans la liste des favoris, il est possible d'insérer une icône de 16 pixels sur 16 sur le site en lui donnant comme nom *favicon.ico*. Pour créer ce genre d'image, il existe un certain nombre de logiciels. IconArt est gratuit, téléchargeable sur `www.conware-pro.com/products/ia`.



◄ Figure 2.19 : *IconArt*

Web Developer

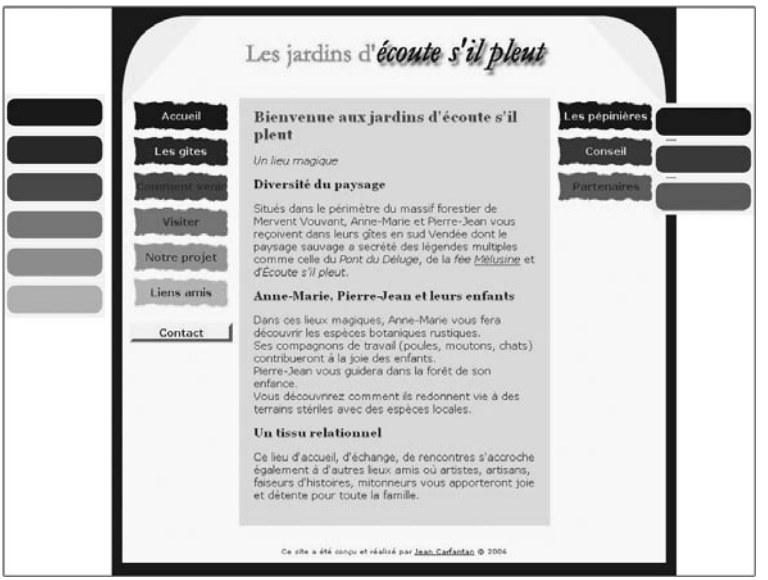
L'extension Web Developer de Firefox (`https://addons.mozilla.org/extensions/moreinfo.php?id=60`) est un outil intéressant et indispensable. Il permet de voir les coulisses des pages web et, entre autres, d'avoir un accès direct aux feuilles de style. Non seulement les styles deviennent visibles mais ils sont, en plus, modifiables, ce qui est intéressant. Cette dernière fonctionnalité est nécessaire car le développeur peut voir comment une page a été conçue, modifier le code et voir instantanément le résultat sur le navigateur. Il suffit de prendre comme étalon un site qui sert de référence et de le modifier. Une fois la feuille modifiée, elle peut être enregistrée. Évidemment, un site ne se construit pas ainsi, mais cette extension peut constituer une base modifiable qui évite de faire tout le code. La seule restriction est que certains sites mal conçus ne sont pas bien affichés sur Firefox.



▲ Figure 2.20 : *Web Developer*

2.9 Créer l'image du site

L'image la plus importante est la bannière qui coiffe les pages. À partir de ses couleurs, il est possible de déterminer les couleurs qui seront employées sur les pages. L'image donne l'impression générale, à travers le fond et la structure de la page, la longueur des textes, l'agencement des boîtes, les filets de couleur, le type d'image. Ensuite le visiteur va découvrir le site. Il s'agit de séduire le visiteur pour qu'il reste puis revienne et participe à l'interactivité du site en se joignant aux forums, en envoyant son opinion ou des suggestions, en achetant des articles en vente dans la boutique. La création du site commence par l'élaboration sur une feuille d'un croquis, puis se poursuit par la création d'un modèle dans un logiciel graphique. Viennent enfin le découpage de l'image et le remplacement de tout ce qui peut être décrit par du code.



▲ Figure 2.21 : *Élaboration*

2.10 Bâtir le site

Une fois la base établie en HTML à partir de l'image originelle, le travail consiste à affiner et à ordonner les différents codes, que ce soit la page elle-même avec le XHTML ou la feuille de style, pour créer des documents qui sont de vrais outils, et non pas un chantier avec des échafaudages encore présents et des tonnes de matériaux en vrac. Le code soit être lisible, non seulement par des machines, mais aussi par des humains, en particulier celui ou celle qui sera chargé de rafraîchir le code au cours des mois suivants. Une documentation sera la bienvenue pour éclaircir les fonctionnalités des différents outils et leur mode d'emploi.

```
/* LES CONTENEURS PRINCIPAUX*/
/*////////////////////////*/
body {
margin:0;
padding:0;
background:#233A00;
font:1em Verdana, Arial, Helvetica, sans-serif;
text-align: center;
}
```

▲ Figure 2.22 : *Code*

Le code XHTML

Grâce à la balise `div` et aux feuilles de style, il est possible de choisir ce qui sera affiché en premier. Le principe du XHTML est celui du flux. Il est donc possible de positionner des boîtes dans le flux, c'est-à-dire selon l'ordre linéaire du code, ou en dehors du flux, d'une manière absolue, au pixel près, ou relative par rapport aux autres boîtes.

La feuille de style

Pour bâtir la feuille de style, le développeur définit en premier lieu les balises les plus génériques, qui vont fournir les propriétés par défaut, puis il s'occupe des détails.

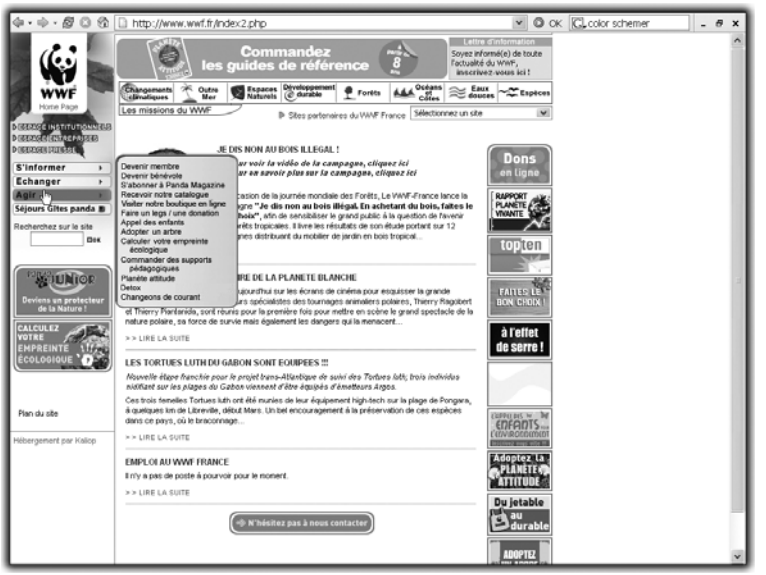
Il commence par définir le sélecteur `body` dont hériteront les autres balises. Puis on déclare les sélecteurs de moindre portée avec leurs propriétés.

```
/* LES CONTENEURS PRINCIPAUX*/
/*//////////////////////////*/
body {
margin:0;
padding:0;
background:#233A00;
font:1em Verdana, Arial, Helvetica, sans-serif;
text-align: center;
}
/* ON ENTRE DANS LA MARINE*/
/*//////////////////////////*/
#navigation {
background:#255899;
font:1.5em;
text-align: left;
}
```

▲ Figure 2.23 : *La feuille de style*

Au fur et à mesure, le développeur sculpte sa feuille de style, c'est-à-dire qu'il donne un aspect général à sa page web, la visionne, puis affine peu à peu au niveau des balises de lien de navigation, rectifie les désordres causés par une ligne de code inappropriée. Ensuite, il construit les classes pour des parties particulières de ses pages ou pour nommer un contenu. Ainsi, il définira le style des pieds de page, avec les propriétés de couleur et les caractéristiques des polices. Pour agir sur un élément et lui donner un comportement, il l'affecte d'un identifiant. Enfin, pour donner du dynamisme et du mouvement, il utilise les pseudo-classes comme `a:hover`, pour que les liens modifient la couleur au passage de la souris. Par la suite, la feuille s'enrichira. Elle pourra aussi se diviser en plusieurs

feuilles pour matérialiser des aspects différents ou s'adresser à des médias différents. Les feuilles de style secondaires seront importées dans la feuille principale.



▲ Figure 2.24 : *Les liens*

Une feuille de style est un document qui se doit d'être lisible pour faciliter les modifications qui ne manqueront pas de survenir.

L'élégance d'une feuille de style consiste en deux aspects :

- bien communiquer sur la feuille par l'insertion judicieuse de commentaires explicites, par des lignes blanches entre différents groupes, par des lignes de moins de 80 caractères ;
- concevoir la feuille avec élégance, c'est-à-dire appliquer des simplifications, des astuces pour court-circuiter du code "bavard".

Pour insérer des commentaires, il suffit de les placer entre /* et */.

```
/*Ceci est appliqué sur toutes les listes de la page*/
li {
font-size: 0.8em;
```

```
}

/*Ceci est uniquement appliqué sur les listes numérotées de notes*/
li.notes {
ol { list-style-type: lower-alpha } /* a b c d e…*/
}
```

La communication se fait à l'adresse du développeur, qui doit se retrouver dans son code plusieurs mois après. La présentation étant séparée du contenu en deux fichiers distincts, il est malaisé de rechercher dans le code source de la page la classe ou l'identifiant. Il s'agit de rendre la tâche plus facile au développeur qui succède au créateur de code.

Les étapes

Pour la construction d'une feuille de style, le développeur fait une sorte de zoom avant, c'est-à-dire qu'il définit d'abord les grandes lignes comme les différentes zones pour lesquelles il utilisera des boîtes. Dans chaque zone, une impression d'ensemble préside, constituée d'un arrière-plan, d'une police de caractères ou parfois d'un interlignage différent. Le site imaginé commence à prendre corps. Le développeur joue avec l'héritage, en définissant des caractéristiques de toute la page, puis surcharge certains sélecteurs et propriétés pour certains détails. La page s'affine.

La surcharge

La surcharge dans les langages objet est basée sur la notion d'héritage. Si une propriété est définie dans un sélecteur comme `body`, tous les sélecteurs enfants de cette balise en hériteront. Ainsi, quand la valeur de la propriété `font-size` est `1em`, toutes les balises `p` bénéficieront à leur tour de cette valeur par défaut. Seule la surcharge de cette propriété permettra de modifier cette valeur en la déclarant dans le sélecteur `p` avec une autre valeur.

Le développeur a déjà défini les différentes parties de la page avec précision. Il sait qu'il y aura des liens de navigation, des liens d'incise (à l'intérieur des textes) et des liens de référence (à la fin d'un texte) et qu'ils auront tous un aspect différent. Alors sa feuille de style va se créer comme une énumération des différents sélecteurs et propriétés. Ensuite, il implémente les différentes classes et identifiants dans ses balises XHTML. Il lui faudra, bien sûr, ordonner ces codes CSS, les présenter, en un mot

les rendre lisibles. Il pourra tester un sélecteur supplémentaire puis l'invalider en le mettant en commentaires. Une fois le code arrêté, il ne reste plus qu'à le nettoyer, le simplifier, voir ce qui peut être enlevé sans provoquer de dysfonctionnement dans la présentation.

```
/* LES CONTENEURS PRINCIPAUX*/
/*////////////////////////*/
body {
margin:0;
padding:0;
background:#233A00;
font:1em Verdana, Arial, Helvetica, sans-serif;
text-align: center;
}
div#contenu {
background:#ffffe1;
background-repeat:no-repeat;
background-position:top left;
background-image:url("images/haut.gif");
max-width: 770px;
margin: 10px auto 0 auto;
text-align: left;
border: 1px solid #233A00;
padding: 0;
}
```

▲ Figure 2.25 : *La surcharge*

La validation

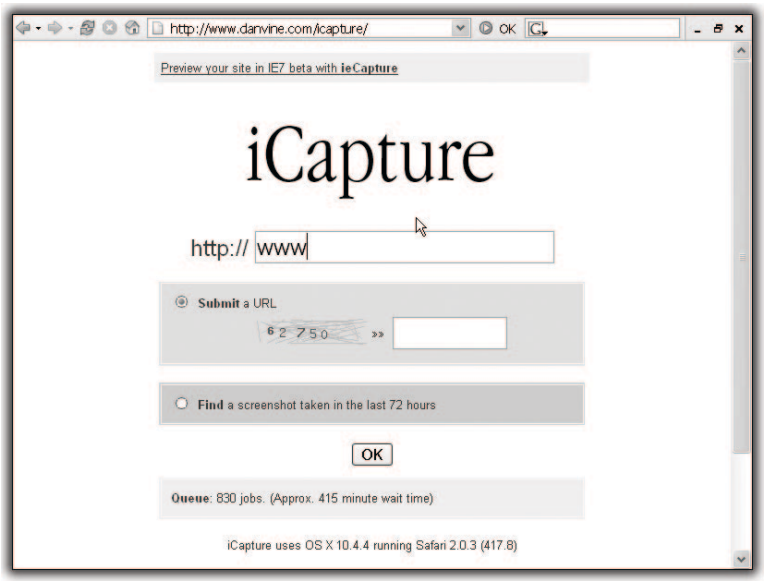
Une fois la feuille de style créée, il est nécessaire de la tester dans tous les navigateurs du marché, ou au moins les plus courants.

Pour vraiment être complet, il faudrait avoir un PC Windows, un Macintosh et un PC Linux afin de tester l'affichage sous différentes systèmes d'exploitation et différents navigateurs et de voir ainsi le site

sous toutes les coutures. Mais avoir plusieurs machines est coûteux pour un individu. Certaines sociétés de développement web en arrivent à cette extrémité.

Les utilisateurs de Macintosh peuvent émuler un PC sur leur machine et donc installer Windows et utiliser tel ou tel navigateur web. Il semble, en revanche, que la puissance est atténuée par rapport à un vrai PC. Cette solution, d'autre part, entraîne des frais.

Il est possible de vérifier un site sur d'autres machines, chez des amis, dans des cybercafés ou d'autres lieux publics, mais il ne s'agit là que d'une phase de vérification, et non de validation pendant laquelle il faut pouvoir modifier le code et voir le résultat en temps réel.



▲ Figure 2.26 : *icapture*

▲ Figure 2.27 : *iecapture*

Les utilisateurs de PC peuvent visionner leurs pages web telles qu'elles s'afficheront sur un "Macintosh", grâce à `www.danvine.com/icapture`. Sur ce même site, sur la page `www.danvine.com/iecapture`, il existe un émulateur pour de nombreux navigateurs, mais cette page est la plupart du temps hors service, sans doute dépassée par son succès.

▲ Figure 2.28 : *browsercam*

Il existe quelques émulateurs payants sur Internet. `www.browsercam.com` en propose un qui peut être utilisé gratuitement pendant 24 heures, pour

faire jusqu'à deux cent captures. Il montre le rendu d'une page sur différents systèmes d'exploitation et navigateurs. C'est encore la solution la plus rationnelle. Il est possible ensuite de valider le code sur le site `http://validator.w3.org/`.

▲ Figure 2.29 : *validator*

La fonctionnalité du site

Il reste à savoir si le site fonctionne bien, c'est-à-dire s'il communique bien, si l'intention première est respectée, si chaque fonctionnalité est clairement perçue, si le principe de navigation est compris, si la lecture est aisée pour tous (catégories socioculturelles, déficients visuels, cultures minoritaires, etc. selon la cible).

Si le site propose des liens non soulignés, ils doivent être assez évidents pour que le visiteur sache qu'il peut cliquer dessus. Si une information est importante, il est peu judicieux de la placer en bas de la page à gauche. Elle serait mieux disposée en haut à gauche ou à hauteur des yeux. Les mots sibyllins associés à des liens ne poussent pas les internautes à visiter la page pointée. Il vaut mieux être explicite et utiliser à dessein la balise

`title` dans les liens. Le manque de contraste entre le texte et le fond de page rend toute lecture difficile car l'œil a besoin d'un bon contraste pour discerner des caractères (affichés en pixels qui plus est). Certaines couleurs ne sont pas vues par les daltoniens. Il vaut mieux éviter qu'elles aient une signification dans les fonctionnalités du site. En d'autres termes, il faut proscrire des liens rouges et des liens verts qui orienteraient les visiteurs, respectivement, sur une page du site ou la page d'un site externe.

◄ Figure 2.30 : *Lien Contact*

Il existe plusieurs moyens de savoir si un site communique bien. Tracer le visiteur au long de sa déambulation, en créant une session et en la stockant, aide à analyser le parcours type et à déterminer les parties du site ignorées. Le sondage peut aussi aider à comprendre les détails à améliorer. La page **Contact** est encore le meilleur moyen d'avoir un écho des visiteurs. Il ne faut pas hésiter à les inviter à faire part de leurs opinions et des difficultés qu'ils rencontrent dans la visite du site. Une phrase dans le pied de page avec un lien e-mail est une bonne idée.

Le bon référencement

Si le site est en XHTML et en CSS, il sera déjà plus facile à référencer. Pour qu'un site soit référencé, il doit réunir plus de cinq pages et avoir de nombreux liens qui mènent à lui, selon Google.

L'inscription

L'inscription se fait directement sur le site des moteurs de recherche. Dès que le site est fini, il faut se rendre sur Google, Yahoo!, etc. et soumettre l'adresse URL créée (voir Figure 2.31).

Les moteurs de recherche régionaux

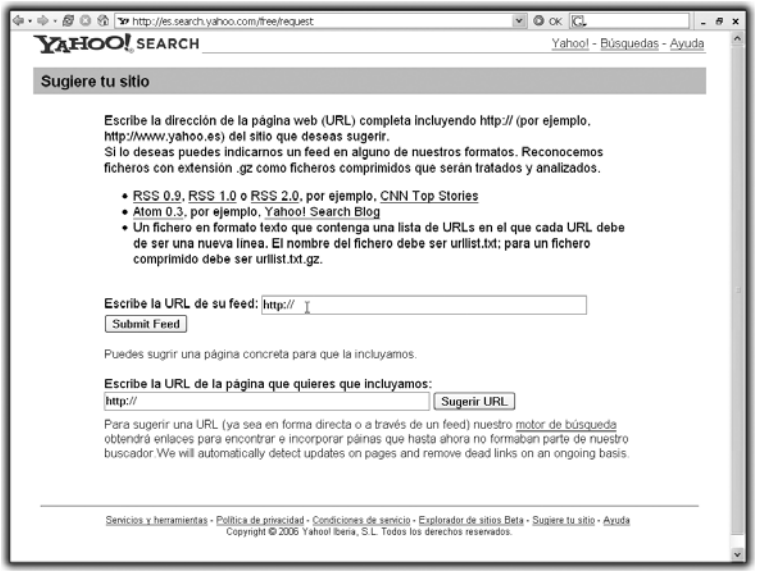
Pour être bien référencé, il vaut mieux se méfier de la diversité des langues prises en charge par les différents moteurs de recherche. Le référencement dans une version n'entraîne pas automatiquement le référencement dans les autres versions du même moteur de recherche.

▲ Figure 2.31 : *Inscription Google*

Tab. 2.1 : Les pages de soumission de site sur les moteurs de recherche

Yahoo!	`http://fr.search.yahoo.com/free/request`
Google	`www.google.fr/addurl/?hl=fr&continue=/addurl`
Voilà	`http://guide-voila.search.ke.voila.fr/Soumission/`
MSN	`http://search.msn.fr/docs/submit.aspx?FORM=WSDD2`

Le site doit être fini et en ligne pour être soumis, sinon le webmaster s'expose à un échec. Une fois l'opération effectuée, il faudra compter entre deux et huit semaines avant que le site n'apparaisse dans l'index.



▲ Figure 2.32 : *Yahoo! Espagne*

Le bon positionnement

Ensuite, le but du webmaster est d'avoir son site bien positionné. Il doit apparaître dans les trois premières pages de résultats d'une recherche sur tel ou tel mot-clé. Le but d'un moteur de recherche est de fournir une réponse exhaustive et pertinente à toute requête de l'internaute afin de fidéliser ce dernier, de s'assurer qu'il revienne faire ses recherches et visualise par la même des encarts de publicité. Il est donc important de bien définir le thème et les sous-thèmes et de les reporter dans les mots-clés et la description du site.

Les balises meta

Voici une liste des nombreuses balises `meta`. Elles ne sont pas toutes indispensables.

- `<title>`*100 caractères maximum (y compris les espaces) sans passage à la ligne*`</title>` ;

- `<meta name="description" content="`*150/200 caractères maximum (y compris les espaces) sans passage à la ligne.*`" />`
- `<meta name="keywords"content="`*1 000 caractères maximum (y compris les espaces). Les 400 premiers seulement sont pris en compte le plus souvent par les moteurs de recherche. 100 mots maximum sinon l'auteur peut être pris pour un spammeur.*`" />`
- `<meta name="abstract" content="`*Bref résumé du site. Peu usité. 200 caractères maximum.*`" />`
- `<meta name="author" content="`*Le nom de l'auteur du site*`" />`
- `<meta name="copyright" content="`*Le nom et la date du copyright du site*`" />`
- `<meta http-equiv="Content-type" content="text/html; charset=iso-8859-1" />`
- `<meta name="language" content="fr" />`
- `<meta http-equiv="Content-Language" content="fr" />`
- `<meta name="robots" content="index, FOLLOW" />`
  - `NOFOLLOW` : ajoute cette page uniquement au moteur de recherche.
  - `NOINDEX` : supprime cette page du moteur de recherche.
  - `All Content` : ajoute toutes les pages au moteur de recherche.
- `<meta name="revisit-after" content="30 days" />`
- `<meta name="reply-to" content="`*webmaster@lesite.com*`" />`
- `<meta name="identifier-url" content="`*http://www.lesite.com/*`" />`
- `<meta name="subject" content="`*Le thème du site*`" />`
- `<meta name="expires" content="never" />`
- `<meta name="publisher" content="`*Nom de l'éditeur du site*`" />`
- `<meta name="location" content="France, FRANCE" />`
- `<meta name="rating" content="general" />`
- `<meta name="category" content="`*une_catégorie*`" />`.

Les balises `meta` les plus utiles sont `keywords` et `description`, qui sont lues par les robots et utilisées par les moteurs pour répondre à une recherche. Le rang dépend de la pertinence, qui se détermine en fonction des mots contenus dans le titre, dans l'URL, dans la page elle-même.

Des générateurs de métatags existent sur Internet (`www.toulouse-renaissance.net/c_outils/c_meta-tag_test.htm` ou `www.referencement-fr.com/generateur_metatags` par exemple).

▲ Figure 2.33 : *Générateur de métatags*

Les mots-clés comme décor

La bonne idée pour être bien référencé est de mettre sur la page une boîte avec des mots-clés, de sorte qu'ils figurent dans la page elle-même. Cette boîte peut être aussi un bon moyen pour rappeler au visiteur les thèmes du site et de la page, comme une illustration. La police et le fond peuvent être suffisamment esthétiques pour servir de décor.

▲ Figure 2.34 : *Boîtes à mots-clés*

Robots.txt

robots.txt est un fichier texte placé dans le répertoire public pour indiquer aux robots qui écument le Web où ils sont autorisés à fureter.

Des sites, comme `http://outils.enaty.com/outils/generateur-robots.php`, existent pour vous aider à créer ce fichier.

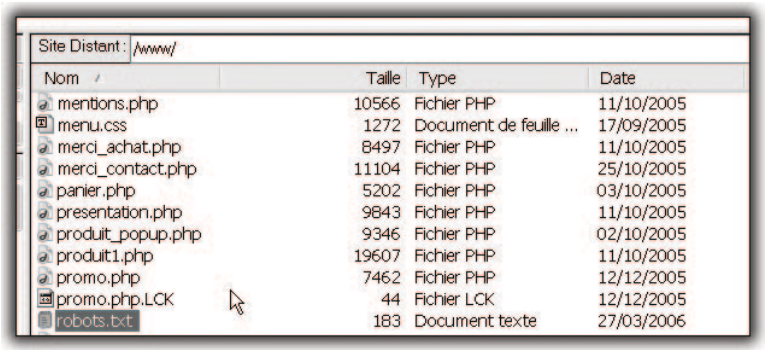
Tab. 2.2 : Exemple de fichier robots.txt

Code	Signification
`User-agent: *`	Permet aux robots de visiter le site.
`Disallow: /secret /`	Interdit aux robots de visiter ce répertoire.
`Disallow: /secrets.htm`	Interdit aux robots de visiter cette page.
`Disallow: /coulisses/index.php`	Interdit aux robots de visiter cette page dans ce répertoire.

Ce fichier *robots.txt* demande aux robots fouineurs de ne pas indexer les répertoires *secret*, *tmp* et le fichier *index.php* dans le répertoire *coulisses*. Chaque fois qu'un robot visite un site, il commence par chercher ce fichier pour savoir ce qui ne doit pas être indexé dans le répertoire web.

▲ Figure 2.35 : *Générateur du fichier robots.txt*

Dans le répertoire secret, mettez les fichiers qui ne regardent personne, mais qui doivent vous être accessibles sur le Web. Aucun lien n'y mène et ils sont sécurisés par le fichier *robots.txt* situé dans le répertoire *www* ou *htdocs* ou *public_html*.



▲ Figure 2.36 : *Arborescence*

Il vaut mieux ne pas y mettre des fichiers trop secrets car certains robots ne respectent pas les commandes de *robots.txt*. Par ailleurs, il n'y a pas que des robots de moteur de recherche sur le Web… *.htaccess* est un fichier beaucoup plus sécurisé pour Apache, mais c'est une autre histoire.

Échanger des liens

Le nombre de liens sur le Web qui mènent à un site lui font gagner des places sur les pages des moteurs de recherche. Il est donc judicieux de créer une page avec des liens vers d'autres sites sur des sujets complémentaires ou sur le même sujet. Si l'un des sites qui proposent des liens vers votre URL est bien référencé, vous le serez aussi par ricochet. Il est aussi intéressant d'inciter les visiteurs à faire un beau lien vers votre site en leur proposant une page avec différentes tailles de bannières et de logos ou des portions de code à insérer sur leur propre site.

2.11 Internet n'a rien inventé

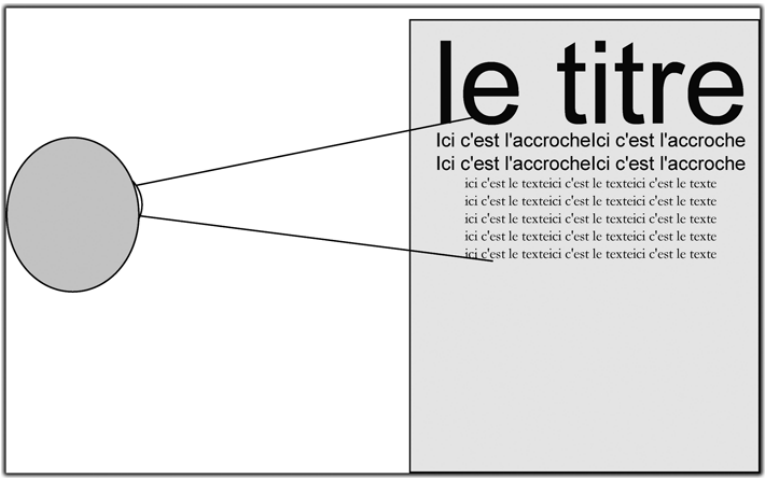
Internet a introduit la rapidité et surtout l'interactivité dans la communication électronique. Mais Internet n'est pas qu'une communication, c'est aussi une expression qui repose sur la culture de l'écrit et sur tous les

progrès réalisés depuis Gutenberg. La manière de lire n'a pas réellement changé car la manière d'écrire n'a pas changé. Quelques aménagements sont nécessaires pour s'adapter à l'écran, mais le substrat d'Internet est le livre et la manière de présenter l'écrit pour le rendre plus lisible, plus compréhensible et plus agréable. Tout l'acquis que l'écrit a eu le temps d'emmagasiner et d'expérimenter par l'imprimerie et la mise en page, jusqu'à la PAO, est à la disposition du créateur de site web. Une bonne connaissance des techniques utilisées dans le passé est un atout pour créer un beau site harmonieux. C'est ce que nous allons étudier dans le prochain chapitre.

3

Les règles de mise en page

Les règles de mise en page, même si elles sont différentes pour l'écran et pour le papier, ont gardé le même socle pour assurer la lisibilité, l'harmonie de l'ensemble et la mise en valeur de certaines parties plus importantes. Le regard humain, quel que soit le support, se pose d'abord aux deux tiers de la page. Il est donc important de connaître quelques bases avant de s'attaquer à la feuille de style elle-même, qui n'est qu'un outil au service d'un savoir-faire.



▲ Figure 3.1 : *Hauteur de la lecture*

3.1 Le texte

Depuis Gutenberg, les caractères d'imprimerie se sont diversifiés. Chaque police de caractères a une personnalité, mais la politique pour une mise en page à l'écran est de rendre le texte lisible tout en veillant à son élégance par rapport à l'ensemble.

Le blanc qui parle

C'est l'équilibre entre la surface imprimée et le support – qui, en typographie, est le papier blanc. Concourent à cet équilibre les polices et les corps, l'interlignage, les gouttières entre les colonnes et les blancs tout autour de la surface imprimée. L'objectif est d'atteindre

une harmonie de proportions entre l'impression et la surface de la page. La première est la masse noire et la seconde est la masse blanche ; elles rivalisent pour donner un gris qui est la dominante de la page vue de loin et qui tire plus ou moins sur le noir ou sur le blanc. C'est le blanc qui donne sa respiration à l'ensemble.

Le français a des règles typographiques différentes de celles de l'anglais. Le rôle de cet ouvrage n'est pas de les passer en revue (d'excellents livres le font), mais simplement de donner quelques bases.

◄ Figure 3.2 : *La une du Monde*

En français, par exemple, il est normal d'accentuer les capitales.

Tab. 3.1 : Accentuation des capitales

Exemple	Question
LES ENFANTS LEGITIMES	LÉGITIMES ou LÉGITIMÉS ?
L'AUGMENTATION DES RETRAITES	Des RETRAITES ou des RETRAITÉS ?

Tab. 3.1 : Accentuation des capitales	
Exemple	Question
LE PALAIS DES CONGRES	CONGRÈS ou CONGRES ?
L'ETUDE DU MODELE	L'ÉTUDE DU MODÈLE ou DU MODELÉ ?

Emploi des mots en capitales

Il est plus malaisé d'accrocher le regard avec une phrase en capitales car les lettres perdent hampes et jambages.

Les lettres sur un écran sont inscrites sur un blanc ou une autre couleur lumineuse et elles apparaissent plus petites que sur du papier.

Les différents types de polices

Sur le Web, les polices sont classées en plusieurs catégories :

Tab. 3.2 : Les différentes catégories de polices	
Nom	Description
Serif	Police dont les caractères ont un empattement (*Times New Roman*).
Sans-serif	Police dont les caractères sont sans empattement (*Arial*).
Cursive	Police imitant l'écriture manuelle (*Monotype Corsiva*).
Fantasy	Police décorative (*Comic Sans MS*).
Monospace	Police dont la chasse est identique quelles que soient les lettres (*Courier New*).

Des polices ont été faites spécialement pour le Web : Tahoma et Verdana (sans-serif), et Georgia (serif).

Pour les autres polices, l'interlignage doit être de 1 ou 2 pixels de plus que la taille de ces polices.

▲ Figure 3.3 : *Hauteur de la lecture*

Tab. 3.3 : Les différentes propriétés de font

Code	Description	Valeur
`font-style`	Définit l'orientation du texte.	`Normal`, `italic`
`font-variant`	Affichage du texte en petites capitales.	`Normal`, `small-caps`
`font-weight`	Définit la graisse du texte.	`Normal`, `bold`, `bolder`, `lighter`, valeur
`font-size`	Définit la taille du texte.	Taille, pourcentage
`line-height`	Définit la hauteur de ligne.	Taille, nombre, pourcentage
`font-stretch`	Définit la largeur des caractères.	`Normal`, `wider`, `narrower`, `ultra-condensed`, `condensed`, `semi-condensed`, `semi-expanded`, `expanded`, `extra-expanded`, `ultra-expanded`
`font-family`	Définit la famille de police à utiliser.	Nom de la police, nom générique d'une police

Taille absolue ou relative

Le choix de l'unité de la taille dépend de la philosophie du webmaster. S'il préfère contrôler sa page, il choisira des unités de mesure absolues. S'il veut accorder plus de libertés à l'utilisateur, les valeurs relatives conviendront parfaitement. Internet Explorer ne permet pas d'ajuster la taille des polices fixes à la vue. Comme Internet Explorer est utilisé par environ 80 % des internautes, il est préférable de recourir à des unités de mesure relatives. L'objectif d'un site web est de pouvoir être vu par le maximum d'internautes ; si le visiteur est intéressé par le thème du site et s'il ne peut

lire le texte parce que la taille de la police est trop petite, il sera frustré et ne reviendra pas.

▲ Figure 3.4 : *Différentes tailles de police*

Tab. 3.4 : Unités de mesure

Taille	Correspondance	Unité	Valeur	Exemple	Type
In	2,54 cm	in	Décimal	2.7in	Absolu
Cm	1/100 de mètre	cm	Entier	4cm	Absolu
Mm	1/1 000 de mètre	mm	Entier	20mm	Absolu
Point	1/72e de pouce	pt	Entier	12pt	Absolu
Pica	12 points	pc	Décimal	2.3pc	Absolu
Pixel	Unité de définition de l'écran	px	Entier	5px	Absolu
Em	Hauteur du M dans la police utilisée	em	Décimal	2.45em	Relatif
Ex	Hauteur du X dans la police utilisée	ex	Décimal	4.5ex	Relatif
%	1 em	%	Entier	110%	Relatif

Comme vous l'avez remarqué, dans les exemples de code, aucun espace ne sépare la valeur de l'unité de mesure.

Les décimales

C'est le point, et non la virgule, qui détermine les décimales.

Tab. 3.5 : Les valeurs nominales

Valeur	Traduction
`xx-small`	Minuscule
`x-small`	Très petit
`small`	Petit
`medium`	Moyen
`large`	Grand
`x-large`	Très grand
`xx-large`	Géant

La taille dépend de la configuration système de l'utilisateur final. Ces différentes tailles conventionnelles sont interprétées différemment selon les navigateurs, la définition et la taille de l'écran et en fonction des polices dont le système dispose. Par exemple, `medium` correspond à 12 pt et `large` à 14.4 pt en théorie. Netscape choisit `small` comme taille par défaut, et Internet Explorer `medium`. Le résultat apparaît donc aléatoire. Si ce système est basé au départ sur une bonne idée, il aboutit en pratique à des mesures approximatives.

▲ Figure 3.5 : *Différentes polices de 24 points*

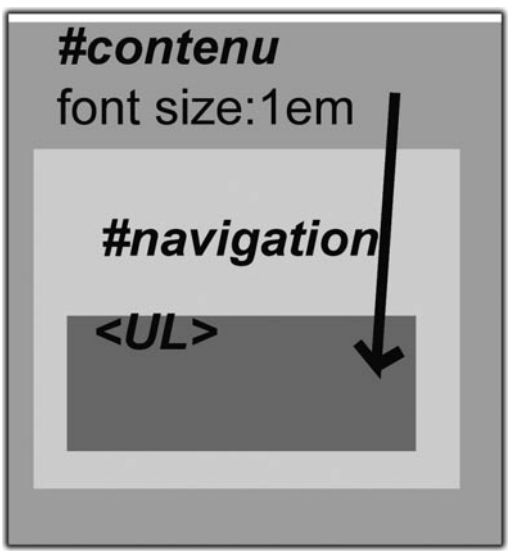
Un exemple de taille relative

Pour s'adapter parfaitement à l'écran d'un ordinateur, il serait idéal de se servir du pixel comme unité car le caractère changerait de taille selon la définition de l'écran. Mais cette option présente deux inconvénients majeurs :

- Internet Explorer ne permet pas d'agrandir le caractère.
- Sur Macintosh, le pixel mesure environ un tiers de plus. L'écran Macintosh est de 72 dpi tandis que celui d'un PC est de 96 dpi, ce qui rend la même police sur Macintosh plus petite.

Les unités de mesure en em ou en pourcentage apparaissent plus adéquates. Il serait possible aussi d'utiliser les mots-clés, mais ils n'autorisent pas la précision que donnent les em ou les pourcentages.

Vous devez considérer la page HTML comme une série de conteneurs. Ils se côtoient ou s'emboîtent comme des boîtes gigognes. Ici nous avons une notation relative. C'est là que l'appellation "en cascade" prend toute son ampleur.



◄ Figure 3.6 : *Héritage*

En fait, un conteneur hérite des propriétés du conteneur qui le contient. Le conteneur qui contient la page et tous ses conteneurs est la balise `body`. Ainsi la propriété `font-size` de `body` sert d'étalon à toute la page et à tous ses conteneurs.

1 em correspond à la taille par défaut des navigateurs, c'est-à-dire à la taille `medium` de la police courante, qui est en général de 16 pixels.

Pour l'affichage d'une police de 10 pixels environ, la propriété `font-size` du sélecteur `body` prend la valeur de 0.625 em, c'est-à-dire la valeur de 10 pixels divisés par 16 pixels.

```
body {font size:0.625em}
```

Avec cette ligne, l'unité em est réglée sur 10 px et désormais tout ce qui est étalonné le sera par rapport à cette mesure. Ainsi 10 px correspondront à 1 em. Ce type de calcul rend plus facile l'attribution des valeurs car 10 est un chiffre rond. Pour 12 pixels, la valeur deviendra 1.2 em, et 1.4 em pour 14 pixels.

Prenons un exemple :

```
<body>
<div id="navigation"> … </div>
<div id="contenu"> … </div>
<div id="pied"> … </div>
</body>

#navigation {font-size:1em}
#contenu {font-size:1.2em}
#pied {font-size:0.9em}
```

Ici, la page type a une colonne de navigation contenant des liens d'une taille de 10 pixels, une colonne principale avec des caractères de 12 pixels et un pied de page avec une police de 9 pixels.

Les paragraphes

Le paragraphe est une vraie unité de texte, avec des propriétés d'alignement, d'alinéa, d'interlignage, etc.

L'alinéa

L'alinéa permet le retrait de la première ligne d'un paragraphe. Cette caractéristique a pour but de rendre visible le changement de paragraphe. Il faut utiliser un espace spécial ou un alinéa pour marquer les différents paragraphes, sinon, dans le cas d'une dernière ligne remplie, le passage n'est pas visible.

```
.paragraphe{
  text-indent:5px;
}
```

La propriété `text-indent` peut prendre comme valeur un nombre positif ou négatif :

- pixel ;
- pourcentage ;
- em ;
- point.

Par défaut, la valeur est `0`.

L'alignement

L'alignement d'un texte dans un paragraphe peut être vertical ou horizontal. Le paragraphe a un fer à gauche ou à droite, ou il peut être justifié, c'est-à-dire aligné des deux côtés. Dans ce dernier cas, la justification joue sur les espacements pour étirer la ligne jusqu'au bord. L'exemple suivant est tiré du site Biblioscript (`www.biblioscript.com/css/txt_vertical.shtml`) (voir Figure 3.7).

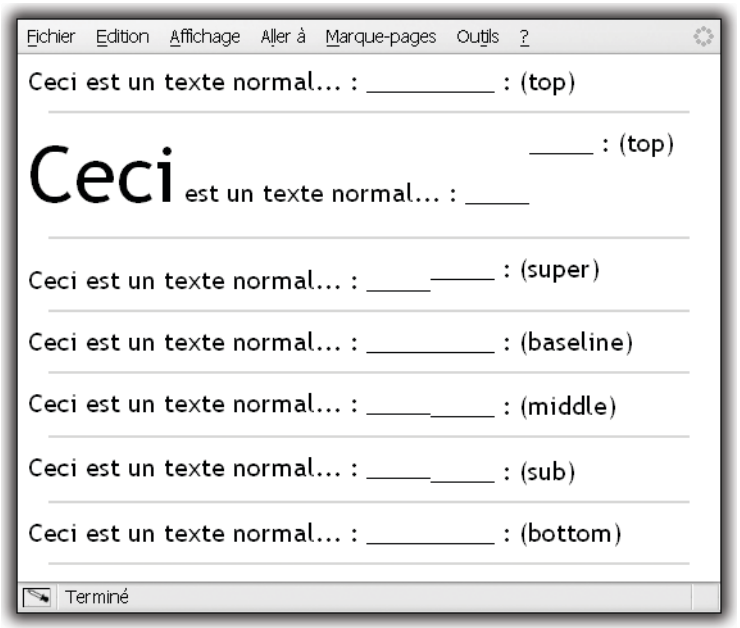
La propriété `vertical-align` concerne les éléments en ligne et la position du contenu à l'intérieur des cellules de tableau. Les éléments en ligne peuvent être des images, le contenu d'une balise `span`, etc.

Tab. 3.6 : L'alignement vertical

Valeur	Effet
`baseline`	Alignement sur la base de la ligne
`middle`	Alignement au milieu de la base de la ligne
`sub`	Alignement en bas de la ligne de base, en indice

Tab. 3.6 : L'alignement vertical

Valeur	Effet
`super`	Alignement au-dessus de la ligne de base, en exposé
`text-top`	Alignement en haut de la boîte de l'élément
`text-bottom`	Alignement en bas de la boîte de l'élément
`top`	Alignement en haut de la boîte de l'élément avec le haut de la boîte de la ligne
`bottom`	Alignement en bas de la boîte de l'élément avec le bas de la boîte de la ligne
La longueur	Alignement au-dessus (valeur positive) ou en dessous (valeur négative) de la ligne de base
Le pourcentage	Alignement au-dessus (valeur positive) ou en dessous (valeur négative) de la ligne de base



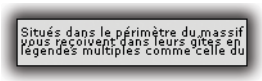
▲ Figure 3.7 : *Vertical-align*

La hauteur de ligne

L'interlignage est la distance entre deux lignes. Pour une bonne clarté, il est conseillé de la porter à 10 % ou 20 % de plus que le corps du caractère. Les unités de base sont les même que celles du caractère.

- pixel ;
- pourcentage ;
- em ;
- point.

```
p {
font-size: 1em;
line-height: 120%;
}
```



◄ Figure 3.8 : *Hauteur de ligne de 0.75 em*

3.2 Les illustrations

Les illustrations accompagnent le texte. En d'autres termes, l'utilisateur voit en premier le texte, puis ensuite les images. Les illustrations sont un complément du contenu du texte pour appuyer ou aider à la compréhension ou encore créer un décalage ironique de sens.

Une bonne illustration doit répondre à certains critères. Elle est au format portrait à la française ou paysage à l'italienne. Pour faire ressortir les illustrations, il est préférable d'avoir des formats variés, qui arrêteront le regard. Pour retenir l'œil, il faut que les personnages et les objets soient bien organisés dans le cadre. Il ne suffit pas de savoir appuyer sur le déclic d'un appareil photo numérique pour s'improviser photographe. Il existe la règle des tiers dans une image. Ainsi, les personnages ou les objets importants doivent se situer à l'intersection des tiers dans l'image. Il s'agit de diviser une photo en trois pour obtenir neuf rectangles. L'œil est conduit vers leur intersection naturellement. La photo utilisée doit être bien contrastée pour être lisible, accrocher le visiteur et avoir un impact.

◄ Figure 3.9 : *Les musées de Poitiers*

Voici quelques questions à se poser sur une illustration :

- Quelle information apporte-t-elle ?
- Décrit-elle une situation ? Est-elle symbolique ?
- Crée-t-elle de l'émotion ?
- Vient-elle en complément du texte ?
- Est-elle un décor dans la page, qui casse la monotonie des blocs de texte ?

◄ Figure 3.10 : *De gauche à droite*

Une image se lit comme un texte. Si elle horizontale, elle se lit de gauche à droite. Les bandes dessinées constituent un excellent objet d'étude. Un personnage, par exemple, qui court de gauche à droite, a l'air d'aller plus

vite que s'il courait de droite à gauche, dans un univers occidental, du fait qu'en Occident, on a l'habitude de lire de gauche à droite.

▲ Figure 3.11 : *Illustrations*

Une illustration se place en position flottante avec la propriété HTML `align`. L'équivalent est la position flottante des boîtes `div`, mais l'image est un élément en ligne, c'est-à-dire qu'elle s'insère dans le flux des balises. En revanche, un sélecteur avec `display: block` est retiré du flux général pour s'insérer dans le flux des blocs et se place donc là où le développeur le décide.

```
img {
  display: block;
  margin: 10px;
  float: left;
}
```

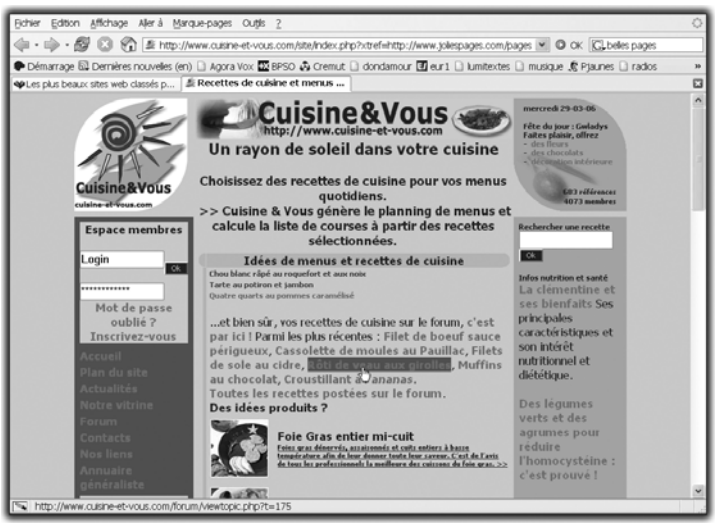
3.3 La disposition

Les CSS permettent de bien maîtriser la position des blocs. Un encadré peut être facilement inséré dans un bloc flottant.

```
div .cadre {
padding: 10px;
margin: 10 px;
background-color: #233A00;
font-size: smaller;
}
```

La stratégie de l'emplacement des blocs vise plusieurs objectifs :

- hiérarchiser l'information ;
- attirer l'attention du lecteur en cassant un agencement trop symétrique et ordonné ;
- casser les gros pavés illisibles en petites unités pour donner plusieurs perspectives sur le même contenu ;
- entraîner le lecteur, par des accroches soigneusement disposées, à lire l'article entier.



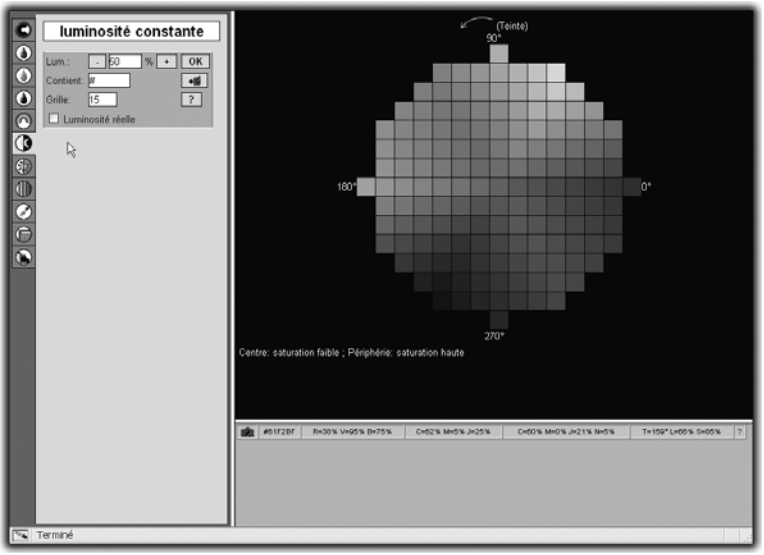
▲ Figure 3.12 : *Disposition des informations*

Un coup d'œil sur certaines pages peut causer une indigestion. La subtilité consiste à donner plusieurs entrées à l'article. Avant la lecture était littérale. Maintenant elle est de plus en plus spatiale. Les téléspectateurs voient chaque chaîne de télévision comme une facette d'un réel morcelé.

Avec la mode des mangas par exemple, les jeunes prennent l'habitude de lire un livre en commençant par la fin, à la façon japonaise. Sur le Web, il en est de même : le visiteur s'introduit dans le site par n'importe quelle page, parcourt, clique, revient, lit encore, comme quand il cherche sur sa télévision le morceau de réel qui va l'intéresser 5 minutes.

3.4 La couleur

La couleur s'exprime de plusieurs façons, via les images, la police de caractères et le fond, qu'il s'agisse de blocs ou de menus, et enfin les filets.

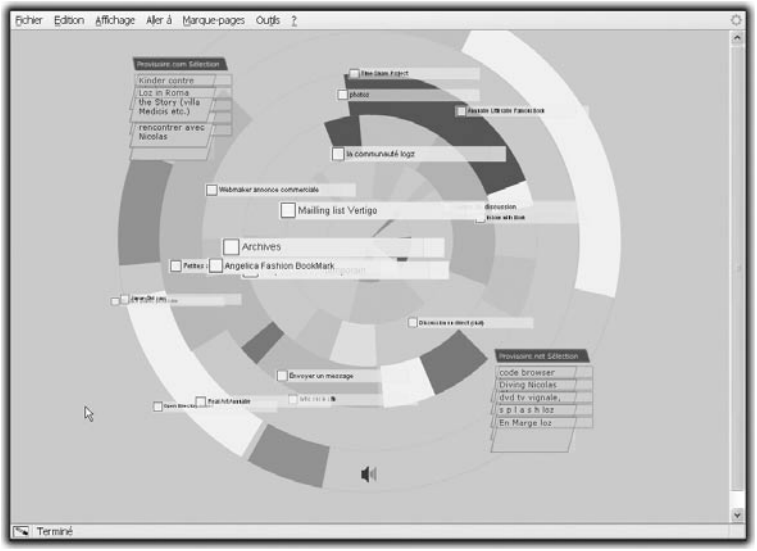


▲ Figure 3.13 : *Le chromographe*

Les couleurs sont restituées à partir des trois couleurs primaires : le rouge, le vert et le bleu (RVB ou RGB). Le mélange des trois donne le blanc en couleurs additives et le noir avec des couleurs soustractives. Les différences de proportion reconstituent les couleurs du spectre. Les couleurs additives sont obtenues avec une source lumineuse comme un spot, un projecteur vidéo ou un écran d'ordinateur (CFT ou LCD). Les couleurs soustractives sont obtenues par le mélange des colorants (procédé utilisé dans l'imprimerie). Comme la somme de tous les pigments ne donne pas

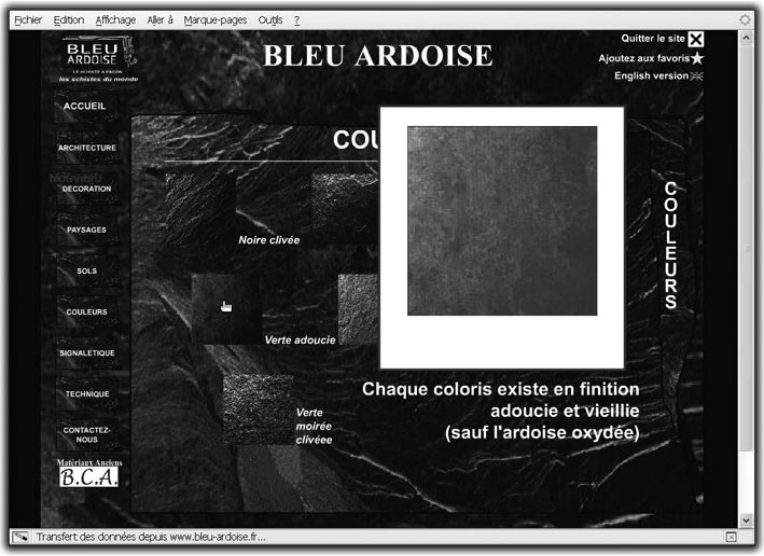
exactement le noir, mais un marron foncé, on a ajouté le noir. Ainsi ce procédé se nomme CMJN (cyan, magenta, jaune et noir). C'est la quadrichromie.

Le contexte coloré de l'ensemble joue aussi un rôle essentiel sur la perception de la couleur d'un objet. Ainsi, une police de caractères rouge ou bleue sur un fond noir paraît plus claire, et plus foncée sur un fond blanc. Ces effets de contraste ainsi que d'autres mécanismes visuels interviennent dans la perception de la couleur.



▲ Figure 3.14 : *http://provisoire.com*

L'œil est influencé par la perception du cerveau. Dans les langues celtiques, la couleur glas (en breton) représente le vert, le bleu et le gris avec, pour référence, l'ardoise. En breton moderne, le mot "gwer" a été ajouté pour caractériser les étoffes vertes (le vert synthétique). Les anciens Bretons distinguaient-ils ce vert ? Cette étude a aussi été faite sur les anciens Grecs. Ainsi le cerveau distingue différemment les couleurs, selon leur contexte et selon la culture du spectateur.



▲ Figure 3.15 : *Couleur ardoise bleu-ardoise.fr*

La balance des blancs dépend de la température de couleur. Elles ne sont pas les mêmes selon le moment du jour : elles passent du rouge au jaune, puis au bleu. Ces variations sont mesurées en kelvins. Les écrans d'ordinateurs sont calibrés différemment selon leur type (autour de 9 300° K pour un CFT, et de 6 500° K pour un TFT) et leur marque. Le calibrage se fait sur un blanc "neutre". C'est l'étalon sur lequel se base tout le spectre affiché à l'écran.

- 9 300° K (blanc bleuté lumineux) ;
- 7 500° K (blanc bleuté) ;
- 6 500° K (blanc légèrement bleuté) ;
- 5 000° K (blanc légèrement jaune) (voir Figure 3.16).

Ainsi le spectre de couleurs que le développeur voudra rendre sur son site variera d'un moniteur à l'autre. Il est donc inutile d'être trop perfectionniste dans ce domaine.

On distingue les couleurs froides, chaudes ou neutres, relaxantes ou toniques, complémentaires ou opposées, etc.

▲ Figure 3.16 : *Balance des blancs*

Le logiciel Color Schemer permet de dégager quelques couleurs dominantes. À celles-ci peut s'ajouter une couleur tonique qui "réveillera" le visiteur.

Il existe deux principales propriétés de couleur :

- `color` concerne la couleur de surface, c'est-à-dire le plus souvent la couleur du texte.
- `background-color` désigne la couleur de fond.

Notation hexadécimale

Les couleurs sont désignées par un code hexadécimal complet (ou abrégé), de 6 caractères précédés par un dièse (#), ce qui permet d'économiser de la place. 16 777 216 couleurs sont ainsi répertoriées.

◄ Figure 3.17 : *Hexa*

Le code hexadécimal des couleurs

Les nombres en Occident sont en base 10 (de 0 à 9). C'est le code décimal. Le code hexadécimal est en base 16. Après 9, il utilise les lettres A pour 10, B pour 11… jusqu'à F pour 15. La couleur est codée sur 8 bits. Chaque couleur de base (rouge, vert, bleu) est codée sur deux caractères (de 16 possibilités chacun), ce qui donne 256 possibilités, donc 216^3, soit 16 777 216 possibilités. Le code couleur de ce qu'on appelle les couleurs sécurisées, référence 216 couleurs.

Les codes formés de paires de caractères identiques peuvent être abrégés. Ainsi `#11BBDD` correspond à `#1BD`.

Notation par mot-clé

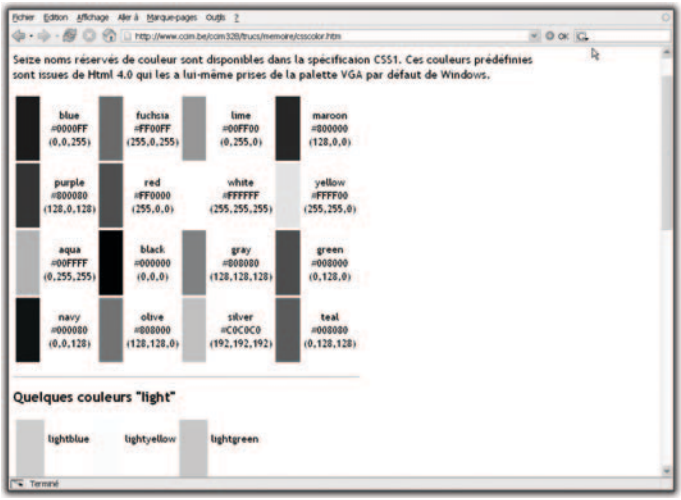
Les couleurs peuvent être nommées par leur nom en anglais :

Tab. 3.7 : Les mots-clés représentant les couleurs

Mot-clé	Correspondance
`aqua`	Cyan
`black`	Noir
`blue`	Bleu moyen
`fuchsia`	Rose fuchsia
`gray`	Gris foncé
`green`	Vert moyen
`lime`	Vert printemps
`maroon`	Marron foncé
`navy`	Bleu marine
`olive`	Vert olive
`purple`	Violet
`red`	Rouge
`silver`	Gris moyen
`teal`	Vert sapin

Tab. 3.7 : Les mots-clés représentant les couleurs

Mot-clé	Correspondance
`white`	Blanc
`yellow`	Jaune



▲ Figure 3.18 : *Couleurs prédéfinies*

Attention aux erreurs d'orthographe en anglais !

```
color: teal;
```

Notation en pourcentage

Cette notation consiste à calibrer de 0 % à 100 % la quantité de chaque composante (rouge, vert, bleu) d'une couleur, 0 % correspondant à l'absence de la composante. Il est possible d'utiliser des décimales, ce qui accroît le nombre de nuances possibles.

```
color: rgb(42%, 70.25%, 0%);
```

Notation décimale

Cette notation utilise les nombres de 0 à 255 pour caractériser la proportion des composantes (rouge, vert et bleu) de la couleur.

```
color: rgb(25, 0, 255);
```

0 dénote l'absence de la composante et 255 la présence à 100 % de la composante.

Notation en référence à l'interface de l'utilisateur

Cette notation donne au développeur les outils pour imiter les fonctionnalités auxquelles l'utilisateur est habitué. Ces couleurs intègrent l'interface du navigateur avec les couleurs du site comme sur le site Gaesys (`www.gaesys.com`).

▲ Figure 3.19 : *Gaesys*

Certaines possibilités, comme la coloration de la barre de défilement du navigateur, ne sont disponibles qu'avec Internet Explorer. Pour intégrer cette glissière verticale, il faut dans ce cas intégrer la couleur de l'ascenseur dans la page.

Tab. 3.8 : Les couleurs se référant à l'interface de l'utilisateur

Désignation	Signification
ActiveBorder	Couleur de la barre de titre active de la fenêtre
ActiveCaption	Couleur du titre de la barre de titre active de la fenêtre
AppWorkspace	Couleur de l'arrière-plan de la fenêtre de l'application active
Background	Couleur de l'arrière-plan du Bureau
ButtonFace	Couleur des boutons des boîtes de message
ButtonHighlight	Couleur des ombrages en 3D des boutons des boîtes de message
ButtonText	Couleur du texte des boutons dans les boîtes de message
CaptionText	Couleur des titres dans les boîtes de message
GrayText	Couleur du texte désactivé dans les boîtes de message
Highlight	Couleur des éléments sélectionnés dans des listes de choix
HighlightText	Couleur du texte sélectionné
InactiveBorder	Couleur d'une ligne de titre d'une fenêtre non active
InactiveCaption	Couleur du titre dans la ligne de titre d'une fenêtre non active
InfoBackground	Couleur de fond des info-bulles
InfoText	Couleur du texte des info-bulles
Mmenu	Couleur des barres de menus
MenuText	Couleur des éléments de menus
Scrollbar	Couleur de la barre de défilement dans les fenêtres
ThreeDDarkshadow	Couleur sombre des ombrages des éléments 3D
ThreeDFace	Couleur des éléments 3D
ThreeDHighlight	Couleur des éléments 3D sur lesquels l'utilisateur vient de cliquer
ThreeDLightShadow	Couleur claire des ombrages des éléments 3D
ThreeDShadow	Couleur sombre des ombrages des éléments 3D
Window	Couleur d'arrière-plan des fenêtres de document
WindowFrame	Couleur des bordures de fenêtre
WindowText	Couleur de texte normal des fenêtres de document

On peut imaginer une fausse fenêtre de message, créée de toute pièce à partir des attributs de couleur courants.

```
.fenetre {
color: buttonText;
}
```

Les couleurs et images d'arrière-plan

La couleur de fond obéit à la propriété `background-color`.

```
body {
color: black;
background-color: #FD9;
background-image: url(images/image_de_fond.png);
background-repeat: no-repeat;
background-position: top left;
}
```

L'image désignée par `background-image` peut être répétée avec `background-repeat` et positionnée avec `background-position`.

▲ Figure 3.20 : *Fond fixe 1 hersandesign.com*

Il est possible de faire en sorte que le décor suive l'ascenseur vertical.

```
background-attachment:fixed;
```

▲ Figure 3.21 : *Fond fixe 2 hersandesign.com*

Les valeurs peuvent être regroupées dans la propriété `background` de la façon suivante :

`background: background-color background-image background-repeat background-attachment background-position;`

```
body {
background: navy url(image_de_fond.png)
no-repeat fixed left bottom;
}
```

Background-position

La propriété `background-position` définit une position verticale et une position horizontale : `top` ou `bottom`, `left` ou `right`. Cette position peut

être aussi un pourcentage de gauche à droite et de haut en bas (de 0 % à 100 %).

```
background-position: 0% 100%;
```

L'équivalent est :

```
background-position: top right;
```

Pour placer un décor au milieu :

```
background-position: top right;
```

L'équivalent est :

```
background-position: 50% 50%;
```

Background-repeat

La valeur par défaut de `background-repeat` est `repeat`.

Tab. 3.9 : La répétition du décor

Valeur	Effet
`no-repeat`	L'image n'est affichée qu'une fois.
`repeat-x`	L'image est répétée horizontalement.
`repeat-y`	L'image est répétée verticalement.
`repeat`	L'image est répétée ad libitum.

3.5 La pagination

La pagination dans un livre est la distribution du texte en pages qui se suivent, tandis que sur le Web, la pagination ne comporte pas d'ordre : chaque page est plutôt un chapitre avec un texte beaucoup plus court que celui d'un livre. Par ailleurs, le visiteur peut entrer dans le site par n'importe quelle page. De cela découlent plusieurs obligations :

- faciliter l'identification de la page ;
- reporter les éléments de navigation sur chaque page ;
- garder une cohérence tout au long des pages du site.

Faciliter l'identification de la page

Il s'agit de donner un titre explicite et de bien caractériser le sujet de la page. Pour indiquer un titre, il existe plusieurs manières de procéder. Il est possible de l'intégrer sur la page en haut au centre via la balise `h1`. La barre ou le menu de navigation peut désigner la page courante par une couleur différente. Le lien peut prendre une autre couleur par le seul fait qu'il ne comporte que l'intitulé qui, alors, n'est pas encadré par la balise <a>.vers une page quelconque.

Reporter les éléments de navigation

Le menu de navigation est une clé du site : il contient les liens vers chaque page, il hiérarchise l'information et lui donne une cohérence. La cohérence est donnée par l'ordre d'affichage des liens, ce qui constitue déjà une hiérarchie, mais aussi par des marqueurs comme des couleurs ou des puces différentes. Il est aussi possible de suggérer quelle est la page suivante, via un lien au bas de la page courante qui conduit vers la page suivante avec un intitulé explicite. Ce lien vers la page suivante est très souvent utilisé sur des sites linéaires, comme les tutoriaux.

▲ Figure 3.22 : *www.bleu-safran.com*

Garder la cohérence

La cohérence d'un site est surtout basée sur sa charte graphique, c'est-à-dire sur l'habillage des pages et la disposition des textes, des titres, des menus, qui font que le visiteur sait immédiatement sur quel site il se trouve. C'est l'identité graphique du site. Une façon de l'identifier est aussi d'insérer une icône dans la barre d'adresse et dans l'onglet dans les navigateurs à onglets, ainsi que dans les favoris. Cette icône placée dans le répertoire principal est nommée *favicon.ico*.

C'est ici qu'intervient la feuille de style à laquelle toutes les pages sont rattachées. La feuille de style donne un cachet au site et aide les internautes à l'identifier.

4

La réalisation du site

4.1 La page d'accueil

La page d'accueil est essentielle. Elle donne une première impression, qui sera primordiale pour l'internaute dans sa décision de poursuivre sa visite.

▲ Figure 4.1 : *Une vraie page d'entrée*

La page d'accueil s'affiche quand l'internaute saisit le nom de domaine du site. Si elle est nommée *index* et qu'elle soit la seule à porter ce nom de fichier, elle s'affiche quand l'internaute saisit `http://ardaee.com` dans cet exemple. L'extension de fichier peut être *.htm* ou *.php*.

Il est important que le visiteur soit directement dans le sujet, sans page intermédiaire sur laquelle cliquer, ou pire, sans avoir à attendre qu'une animation se termine ou à chercher un lien trop discret sur lequel cliquer.

La page d'accueil garde la même configuration, ou au moins un même aspect graphique, que le reste du site. Le logo peut être plus gros que sur le reste des pages, pour signifier l'importance de cette page. Si une bannière de publicité apparaît en haut, il vaut mieux faire en sorte que

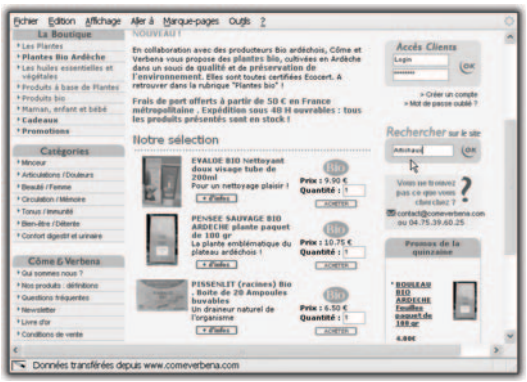
cette publicité ne puisse pas être confondue avec le titre de la page ou prise à tort pour le thème du site.

Lien vers la page d'accueil

Si la barre des liens est identique sur chaque page, il est préférable de ne pas avoir un lien actif vers la page d'accueil sur ladite page d'accueil et de distinguer le lien inactif par une couleur qui le démarque des autres liens.

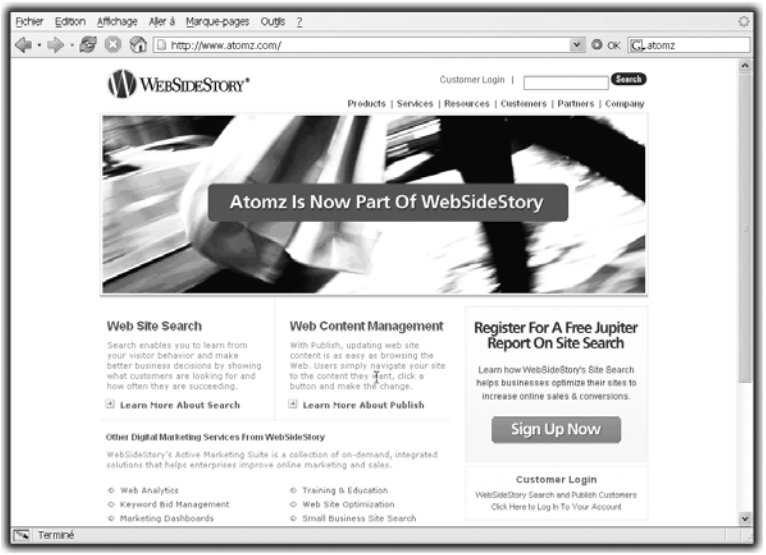
La page d'accueil contient les liens de navigation vers tout le site, l'information essentielle qu'il propose et les dernières nouvelles. En entrant dans la page d'accueil, le visiteur doit découvrir le sujet du site et les thèmes qui seront abordés dans ses pages. Son regard ne doit pas avoir à chercher où aller. Soit le site est un site d'information et il affiche les nouvelles dans sa page d'accueil, soit c'est un autre type de site et les liens de navigation doivent être mis en évidence. En effet, si le nouveau venu s'attarde un peu sur la page d'accueil, l'internaute fidèle, lui, prend la page d'accueil comme un carrefour, un accès central aux autres pages et aux annonces de mise à jour.

"*Que vient chercher l'internaute sur le site ?*" est la bonne question à se poser. Il doit trouver aisément ce qu'il vient chercher en cliquant le moins de fois possible sur des liens.



▲ Figure 4.2 : *Le moteur de recherche comeverbena.com*

Certains visiteurs préfèrent utiliser un moteur de recherche pour naviguer sur un site. Il est donc indispensable d'en installer un en première page. Le moteur de recherche Atomz est gratuit dans sa version basique (`www.osinet.fr/code/prix.asp?CodeMarque=ATOMZ`) et s'insère parfaitement dans un site, en en prenant les couleurs.



▲ Figure 4.3 : *Atomz*

Le titre de cette page donne un premier indice du contenu du site. Entre les balises `title`, il vaut mieux éviter d'inscrire `"Page d'accueil d'Écoute S'il Pleut"` et préférer `"Écoute S'il Pleut, Invitez-vous au jardin extraordinaire !"` par exemple, plutôt que `"Bienvenue sur notre page d'accueil"`.

Sur la page d'accueil, le visiteur est à la croisée des chemins. Plusieurs options peuvent être choisies :

- soit afficher les liens avec chaque thème résumé par un petit paragraphe ;
- soit afficher simplement les liens comme sur n'importe quelle page ;

- soit faire un texte unique qui introduit certaines rubriques plus importantes.

▲ Figure 4.4 : *www.clic-images.com*

Le texte de la page d'accueil doit séduire pour donner envie d'aller plus loin. Même si le visiteur a envie de connaître le thème traité par le site, son désir a besoin d'être attisé. Il faut instaurer ici la confiance, voire la connivence. Le visiteur vient sur la page d'accueil, attiré par un intitulé dans le moteur de recherche ou un lien sur un autre site. Souvent, il ouvre le lien en affichant une nouvelle fenêtre ou en se disant qu'il pourra toujours revenir en arrière avec le bouton adéquat sur son navigateur. Ainsi il parcourt rapidement la page d'accueil pour vérifier si le site satisfait sa curiosité. Il est donc nécessaire de l'accrocher par ce qui fait l'originalité du site. Cette information originale doit être mise en avant, à un endroit stratégique de la page.

4.2 Le site

La création du site a commencé par la création du logo, composé d'un fond jaune clair (`#FFFFE1`) sur lequel figure la phrase "`Les jardins d'Écoute S'il Pleut`", dont la première partie est colorée en vert clair

(#8CBD23) et la deuxième en bleu foncé (#1D1779) avec une ombre portée. La carte du site a été dressée, ce qui permet de monter les liens de navigation. Deux types de liens de navigation se sont dégagés : ceux concernant le gîte et le tourisme et ceux tournés vers le commerce et le service. Il est indispensable de matérialiser leur différence. Le choix des couleurs se porte vers un dégradé de verts : d'un côté des verts à dominante jaune, de l'autre des verts à dominante bleue.

▲ Figure 4.5 : *La bannière ardaee.com*

La construction d'un site commence par la rédaction d'un cahier des charges qui répertorie les desiderata du client et l'aide à préciser sa demande. Ensuite le graphiste crée une page type sur un logiciel graphique. La création des feuilles de style vient ensuite.

Dans notre cas, il est décidé de mettre la totalité des liens de navigation sur chaque page de façon à ce qu'un visiteur atterrissant sur une page quelconque du site ait les outils pour remonter à la page d'accueil ou évoluer directement dans le site.

4.3 Le XHTML

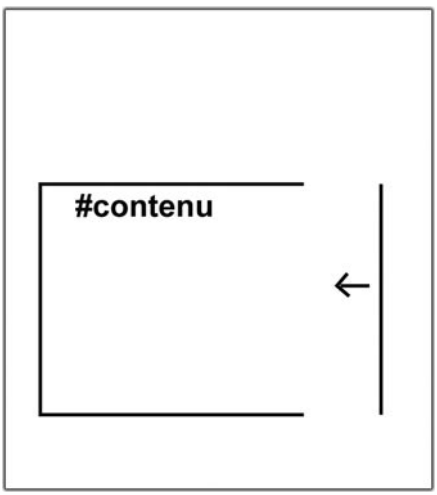
Le XHTML est un HTML plus strict au niveau de ses règles. Les règles sont assez simples à énumérer et des programmes de validation permettent de détecter les erreurs de syntaxe (voir Figure 4.6).

- Toute balise ouvrante doit comporter une balise fermante ou un trait oblique signalant que cette balise est ouvrante et fermante.
- Les containeurs peuvent s'imbriquer, mais pas s'entrelacer.
- Les noms des balises et des propriétés doivent être en minuscules.
- Les propriétés ou attributs doivent avoir une valeur.
- Les valeurs des propriétés sont encadrées par des guillemets.
- Le standard utilisé pour le code de la page doit être précisé.

▲ Figure 4.6 : *XHTML*

La balise fermante

Tout balise ouvrante doit s'accompagner d'une balise fermante.



▲ Figure 4.7 : *Balise à fermer*

Ainsi le code suivant est correct en HTML, mais erroné en XHTML :

```
<p>Ceci est une ligne
<p>Ceci est une deuxième ligne
```

Le paragraphe commençant avec la balise doit se clore par pour être correct sous XHTML :

```
<p>Ceci est une ligne</p>
<p>Ceci est une deuxième ligne</p>
```

De la même façon, le code suivant ne sera pas accepté :

```
Ceci est une ligne<br>
Ceci est une deuxième ligne
```

Le code qui suit sera validé :

```
Ceci est une ligne<br />
Ceci est une deuxième ligne
```

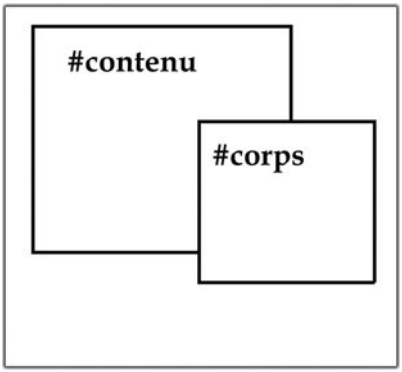
Le trait oblique de fin indique à l'analyseur du navigateur de ne pas chercher une balise fermante. L'espace présent avant la barre oblique assure la compatibilité avec les anciennes versions des navigateurs.

Tab. 4.1 : Les balises solitaires du HTML en XHTML

Balise	Code
IMG	
OPTION	<option …> … </option>
HR	<hr… />
BR	
LI	<li … > …
INPUT	<input … />
DT	<dt … > … </dt>
DD	<dd … > … </dd>

L'imbrication des balises

Les conteneurs s'imbriquent comme des poupées russes.

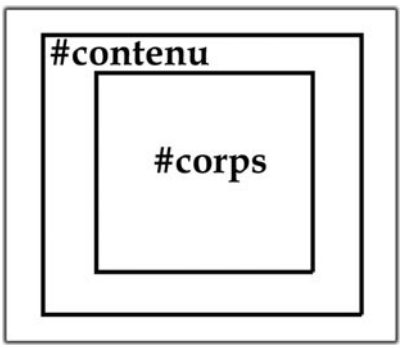


◄ Figure 4.8 : *Conteneurs qui se chevauchent*

Le chevauchement, par contre, n'est pas permis alors que les navigateurs HTML s'en accommodent et le corrigent :

```
<b><u>mauvaise écriture des balises</b></u>
<u><b>écriture correcte des balises</b></u>
<b><u>écriture correcte des balises</u></b>
```

Un conteneur peut contenir un autre conteneur ou être enfermé dans un autre conteneur.



◄ Figure 4.9 : *Conteneur et contenant*

Les dénominations en minuscules

Les noms des balises ou des propriétés doivent être en minuscules.

Invalide :

```
<IMG src="http://monsite.com/image.gif" />
```

Invalide :

```
<img SRC="http://monsite.com/image.gif" />
```

Valide :

```
<img src="http://monsite.com/image.gif" />
```

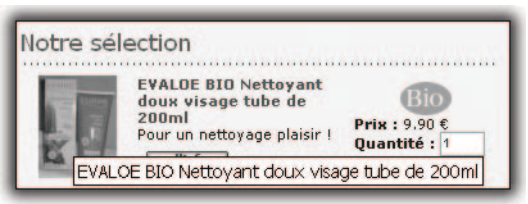
Les propriétés définies

En XHTML, il n'est pas possible de laisser une propriété vide. Ainsi le code suivant est invalide :

```
<img src="http://monsite.com/image.gif" alt=""/>
```

Par ailleurs, il est impossible de mettre une valeur tenant sur deux lignes.

```
<img src="http://monsite.com/image.gif" alt="Mon
portrait"/>
```



▲ Figure 4.10 : *Alt*

Le code suivant est valide :

```
<img src="http://monsite.com/image.gif" alt="Mon portrait" />
```

Certaines propriétés en HTML sont glissées dans les balises sans valeur ni guillemet et en capitales. Le XHTML exige une valeur. Même si cela peut paraître une répétition, ces propriétés prendront comme valeur leur propre nom :

Tab. 4.2 : Les propriétés sans valeur en HTML

Propriété	Valeur
compact	compact="compact"
checked	checked="checked"
declare	declare="declare"
readonly	readonly="readonly"
disabled	disabled="disabled"
selected	selected="selected"
defer	defer="defer"
ismap	ismap="ismap"
nohref	nohref="nohref"
noshade	noshade="noshade"
nowrap	nowrap="nowrap"
multiple	multiple="multiple"
noresize	noresize="noresize"

Le standard

La balise DOCTYPE qui se réfère à une DTD qui est un document qui décrit les directives de syntaxe pour le code qui sera contenu dans la page. Si la page obéit à la norme XHTML 1.0, elle peut avoir une DTD *Transitional*, *Frameset* ou *Strict*. La DTD *Frameset* intervient quand la page contient des trames, c'est-à-dire quand la page est divisé en fenêtres. La DTD *Transitional* permet de mettre à jour plus facilement un site en HTML 4. Elle est moins exigeante que la *Strict* et permet l'utilisation de balises qui ne sont pas dans la DTD, comme la balise center.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//FR"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//FR"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//FR"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

La règle stricte ne permet pas l'usage de certaines balises, comme `dir`, `menu`, `center`, `isindex`, `applet`, `font`, `basefont`, `s`, `strike`, `u`, `iframe` et `noframes`. Certaines propriétés sont aussi refusées. La norme XHTML 1.1 est uniquement stricte et ne comporte pas d'autres DTD.

```
<!DOCTYPE  html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
✂ "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

4.4 Les CSS

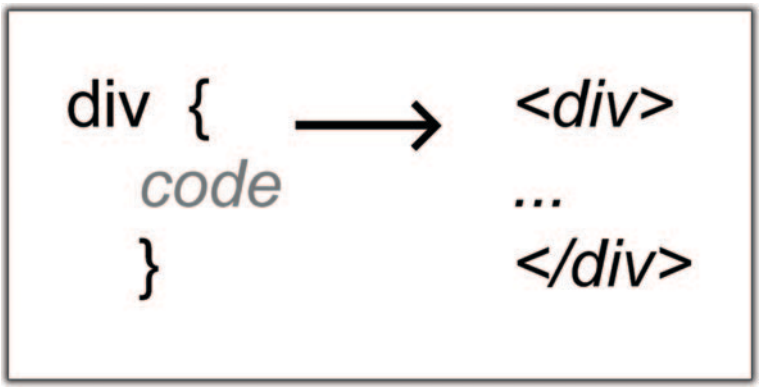
Les feuilles de style CSS sont formées de code séparé du code HTML et qui en précise le format. Pour pointer vers des parties de code HTML, les feuilles de style se servent de sélecteurs, de classes, d'identificateurs et de pseudo-classes.

Les sélecteurs

Les sélecteurs désignent les éléments sur lesquels sont appliquées les règles.

Un sélecteur comprend l'entité désignée suivie d'un espace et d'une accolade ouvrante accompagnée d'une accolade fermante, qui délimitent une ou plusieurs instructions (voir Figure 4.11).

Les instructions sont composées du nom de la propriété suivie de deux points, d'un espace puis de la valeur donnée à cette propriété. Auparavant, il était d'usage de mettre ces indications dans les balises sous forme de propriétés, mais le code était ainsi disséminé.



▲ Figure 4.11 : *Sélecteurs*

Les sélecteurs de type

Les sélecteurs de type pointent vers des conteneurs délimités par des balises simples. Ils contiennent les instructions qui indiquent au contenu d'une balise comment il doit se comporter.

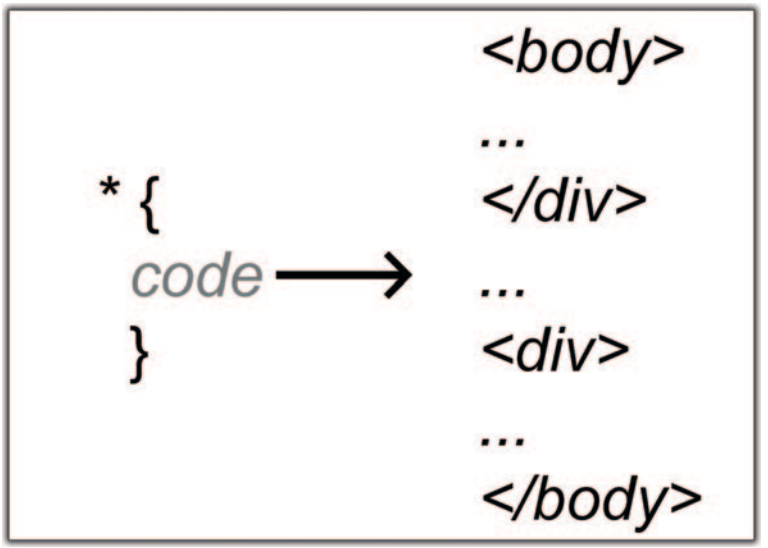
```
h1 {
font-size: 2em;
}
```

Dans la page concernée, tout le texte inclus dans les balises `h1` sera d'une taille de 2 em.

Le sélecteur universel

Le sélecteur universel est peu employé. Il permet de donner des instructions à toutes les balises de la page. On lui préfère souvent le sélecteur `body`, qui remplit le même office (voir Figure 4.12).

```
* {
color: #FF0;
}
```



▲ Figure 4.12 : *Sélecteur universel*

Abréger le code couleur

`#FF0` correspond à la couleur rouge. C'est la forme abrégée de `#FF0000`. Il est courant de voir le code couleur comme trois paires de codes et d'identifier chaque paire par son code. Cela n'est possible que pour les couleurs principales. Ainsi `#9A9` correspond à un gris vert et à `#99AA99`. En revanche, une couleur saumon comme `#FF8A4D` ne peut être abrégée.

Les sélecteurs descendants

Quand le code désigne les descendants d'une balise donnée, le sélecteur ascendant est suivi d'un espace et du sélecteur descendant. Prenons comme exemple le code HTML suivant :

```
<body>
<p>Voici des exemples <ul>
<li>le chat est <strong>blanc</strong></li>
```

```
<li>le ciel est <strong>bleu</strong></li>
<li>l'arbre est <strong>vert</strong></li>
</ul>
<p>Ce sont de <strong>bons</strong> exemples</p>
</body>
```

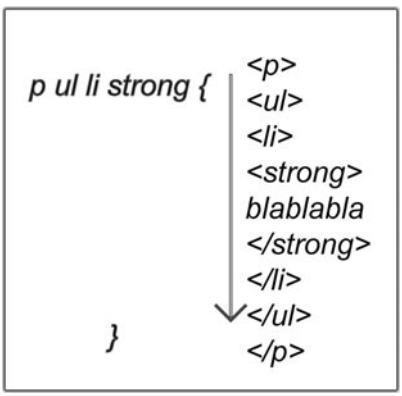
Ce code HTML est enrichi par le code CSS qui suit :

```
p ul li strong {
font-size: 1.5em;
}
```

Le résultat sera que les caractères seront plus grands uniquement dans la liste non ordonnée qui renferme les descendants de la balise `li`, elle-même descendante de la balise `ul`, elle-même descendante de la balise `p`.

les nœuds textes

Le contenu texte d'une balise est son descendant. Voir le DOM au chapitre 9.



◄ Figure 4.13 : *Sélecteur descendant*

Les classes

La classe est une autre manière de pointer vers une partie d'un document. Elle permet aussi de donner une signification à un contenu, en le

dénommant à travers la propriété `class`, à la manière du XML. Cette notion de classe permet au code d'être plus spécifique et d'appliquer du code à des objets apparentés.

```
<body>
<p>Voici des exemples <ul>
<li>le chat est <strong class="blanc">blanc</strong></li>
<li>le ciel est <strong class="bleu">bleu</strong></li>
<li>l'arbre est <strong class="vert">vert</strong></li>
</ul>
<p>Ce sont de <strong>bons</strong> exemples</p>
</body>
```

La feuille de style est codée ainsi :

```
li strong.blanc {
color: #000;
}
li strong.bleu {
color: #27F;
}
li strong.vert {
color: #291;
}
```

Dans chaque balise dont l'attribut `class` a comme valeur `vert`, le texte est coloré en vert, à condition que `class` soit placé dans une balise `strong` et `li`.

Il est aussi possible d'être générique en écrivant ceci :

```
.vert {
color: #291;
}
```

Cette notation marque en vert tous les textes dont la balise contient la classe `vert`.

Les identifiants

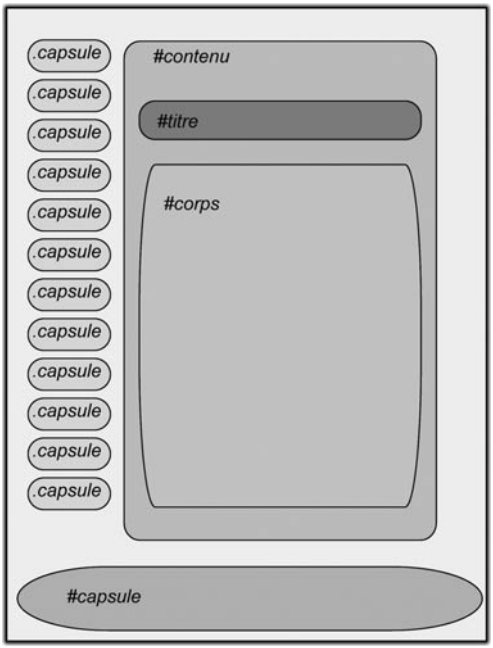
Les identifiants ont le même fonctionnement que les classes, à ce ceci près qu'ils sont uniques. Un identifiant ne peut être utilisé qu'une seule fois dans un document, mais plusieurs fois sur le même site. Il désigne le contenu d'une balise précise.

```
<p id="présentation">Le code présenté ici est doté
d'un identifiant unique</p>
```

Le code CSS :

```
#presentation {
font-face: sans-serif;
font-weight: bold;
color: #291;
}
```

Le texte de cette présentation sera en police sans empattements, en gras et de couleur verte.



◄ Figure 4.14 : *Différence entre classe et identifiant*

L'héritage

Quand les propriétés d'une balise sont spécifiées dans une feuille de style, toutes les balises contenues dans le conteneur en héritent, à moins que l'on ne donne une autre valeur à cette propriété pour l'enfant. Cette dernière

fonctionnalité est parente de la surcharge de propriété dans les classes en programmation objet.

L'héritage permet d'éviter de répéter chaque propriété pour chaque sélecteur. Ainsi si l'on caractérise le sélecteur `body`, toute la page est dotée de la même propriété.

DOM

L'arbre du DOM figure une sorte d'arbre généalogique où une balise contient d'autres balises qui sont ses enfants, ce qui crée des frères, des descendants, etc. . Voir le DOM au chapitre 9.

Les sélecteurs d'enfant

L'héritage fonctionne sur le même mode pour les sélecteurs et les conteneurs. Si un conteneur limité par une balise ouvrante et une balise fermante contient un autre conteneur, ce conteneur prend naturellement les propriétés de son conteneur parent.

```
<b><i>C'est de l'italique qui est gras</i>
et ici c'est juste du gras</b>
```

Si, par exemple le sélecteur `p` reçoit des propriétés, il peut être écrit de deux façons dans la feuille de style. Ici les contenus texte de … seront en bleu :

```
p {
color: blue;
}
```

ou encore :

```
body p {
color: blue;
}
```

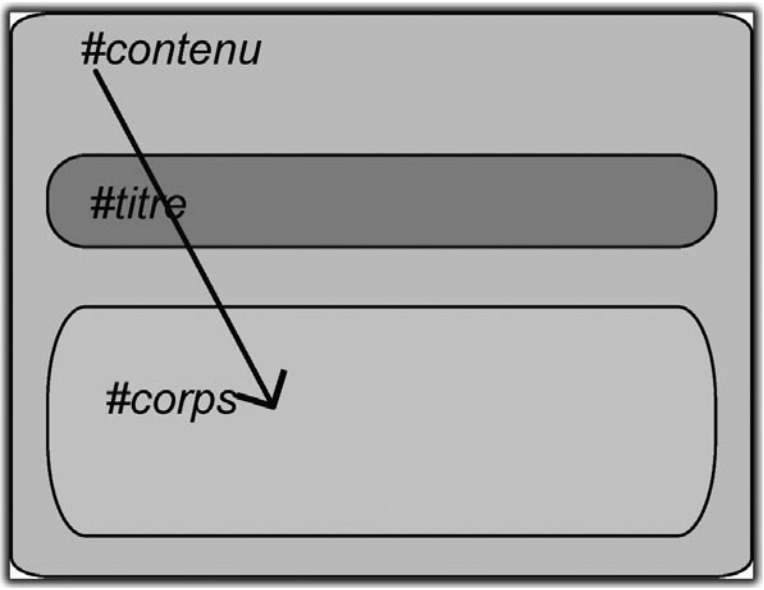
Il est aussi possible de désigner seulement certains descendants d'un sélecteur, comme dans ce code HTML :

```
<body>
<div> <p>Achetez-le immédiatement pour bénéficier
du rabais !</p></div>
<p> Ce logiciel est un couteau suisse pour le son</p>
<p>Avec ce logiciel vous enregistrez tout son qui est émis
par votre carte son !</p>
</body>
```

Le code précédent est accompagné par le code CSS qui suit :

```
body>p {
color: blue;
}
```

Le signe "plus grand que" (>) indique que seules les `p`, enfants directs de `body`, seront concernés par cette instruction. Ainsi le conteneur `p` dans le conteneur `div` sera épargné. Par contre, les conteneurs `p` qui n'ont pas d'autre parent que `body` auront leur texte en bleu.



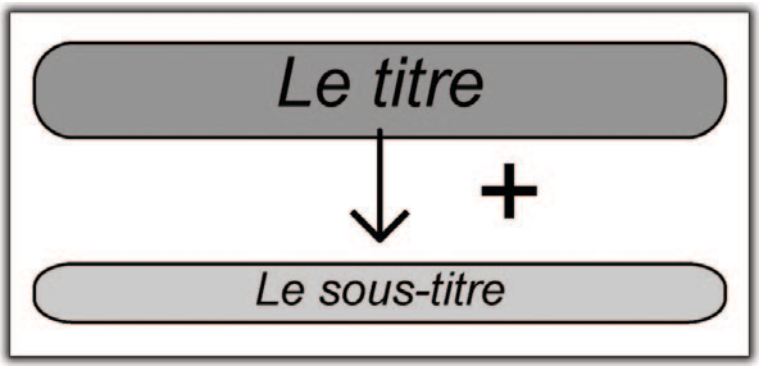
▲ Figure 4.15 : *Sélecteur enfant*

Les sélecteurs d'enfant adjacents

Les sélecteurs d'enfant adjacents sont signifiés par le signe + entre deux sélecteurs à condition que le premier précède le deuxième et qu'ils soient enfants du même parent. Cette fonctionnalité est particulièrement intéressante pour des propriétés qui sont gérées par défaut par le navigateur comme un titre 1 suivi d'un titre 2 qui génère un grand espacement.

```
h1 + h2 { margin-top: -5mm; }
```

Il est possible aussi de modifier l'espace vertical entre une balise p et une balise ul quand le formatage par défaut ne convient pas.



▲ Figure 4.16 : *Enfants adjacents*

Les sélecteurs d'attribut

Les sélecteurs d'attribut permettent de désigner des conteneurs spécifiques en fonction de leurs propriétés ou attributs. Cette fonctionnalité ouvre des perspectives. Par exemple, il est possible de modifier la couleur des liens qui vont vers l'extérieur du site :

```
<p><a href="http://www.photo-de-classe.com">Site gratuit de
➥< retrouvailles</a><br />
<p><a href="liens.php">Tous les liens</a>
```

Un des liens va vers une page extérieure du site, l'autre conduit à un annuaire de liens dans le site.

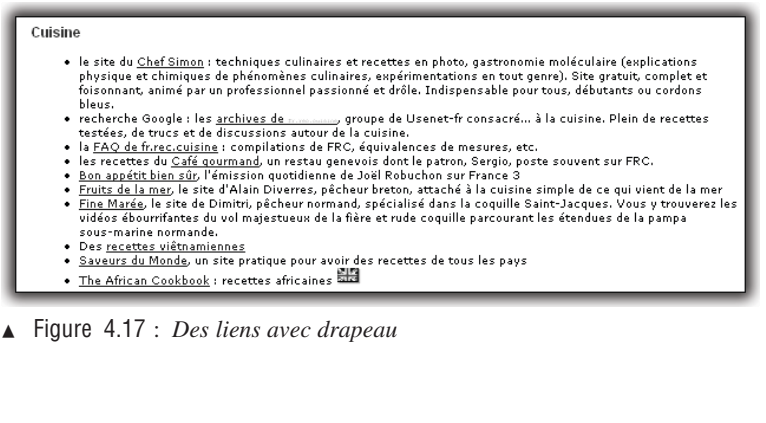
```
a {
color:#029;
}
a[href|="http"] {
color:#261;
}
```

La couleur des liens par défaut sera bleue, et verte pour les liens à l'extérieur.

Une autre possibilité est de marquer les liens qui ouvrent une nouvelle fenêtre dans le navigateur :

```
a {
color:#029;
}
a[target="_blank"] {
color:#261;
}
```

Pour cela, la feuille de style pointe vers les balises des liens qui contiennent la propriété `target="_blank"`. Un autre trait intéressant est d'accoler un drapeau sur les liens qui pointent vers un site en langue étrangère.



▲ Figure 4.17 : *Des liens avec drapeau*

Tab. 4.3 : Exemples de granulation des sélecteurs d'attribut

Sélecteur	Portée	Sélection	Non sélection
`p[lang]`	Tous les paragraphes avec une propriété `lang`	`<p lang="es">`	`<p class="lang">`
`p[lang="fr"]`	Tous les paragraphes dont la propriété `lang` a `fr` comme valeur exacte	`<p lang="fr">`	`<p lang="fr-BE">`
`p[lang~="fr"]`	Tous les paragraphes dont la propriété `lang` est soit le mot `fr` ou qui contient le mot `fr` dans une suite de valeurs	`<p lang="fr br cy">`	`<p lang="fr-BE">`
`p[lang^="fr"]`	Tous les paragraphes dont la propriété `lang` commence par le mot `fr`	`<p lang="fr-BE">`	`<p lang="cy br fr">`
`p[lang$="fr"]`	Tous les paragraphes dont la propriété `lang` finit par le mot `fr`	`<p lang="cy br fr">`	`<p lang="fr-BE">`
`p[lang*="fr"]`	Tous les paragraphes dont la propriété `lang` contient le mot `fr`	`<p lang="fr-BE">`	`<p lang="cy br">`
`p[lang|="fr"]`	Tous les paragraphes dont la propriété `lang` est exactement `fr` ou commence par `fr-`	`<p lang="fr-BE">`	`<p lang="en fr">`

attribut `lang`

L'attribut langue donne comme code le code de la langue en deux lettres parfois suivi d'un tiret et du code d'un pays, ainsi `fr-CA` signifie le français du Canada. Quand il y a plusieurs langues comme `cy br fr`, on peut en faire une énumération séparée par un espace – ici gallois, breton et français.

Les déclarations multiples

Les déclarations peuvent contenir plusieurs lignes. Ainsi une CSS comme celle-ci est correcte :

```
p#surlignement { color: blue; }
p#surlignement { font-size: 1em; }
p#surlignement { background-color: yellow; }
p#surlignement { font-weight: bold; }
p#surlignement { letter-spacing: .3em; }
```

Il est cependant plus simple de regrouper les propriétés dans une seule paire d'accolades pour le même sélecteur :

```
p#surlignement {
color: blue;
font-size: 1em;
background-color: yellow;
font-weight: bold;
letter-spacing: .3em;
}
```

Le regroupement

Le regroupement consiste à réunir des sélecteurs qui ont la même propriété entre les mêmes accolades au lieu de répéter plusieurs fois les propriétés.

```
H1 {
   font-family: Georgia, "Times New Roman", Times, serif;
   font-size: 1.10em;
```

```
   font-weight: normal;
   text-transform: uppercase;
}
h2 {
   font-family: Georgia, "Times New Roman", Times, serif;
   font-size: 1.10em;
   font-weight: normal;
   text-transform: uppercase;
}
h3 {
   font-family: Georgia, "Times New Roman", Times, serif;
   font-size: 1.10em;
   font-weight: normal;
   text-transform: uppercase;
}
```

Ces propriétés sont regroupées comme ceci :

```
h1 h2 h3 {
   font-family: Georgia, "Times New Roman", Times, serif;
   font-size: 1.10em;
   font-weight: normal;
   text-transform: uppercase;
}
```

Les fonctions qui simplifient le code allègent le document CSS et la bande passante consommée.

La bande passante

La bande passante est le débit consommé par un site. Ce débit est composé de tous les fichiers qui sont envoyés sur le réseau suite à une requête. Les hébergeurs donnent un forfait de bande passante pour chaque type de site et les dépassements sont facturés.

4.5 L'habillage de la page

Au début de ce chapitre, nous avons vu un exemple de balise `body`. Vous pouvez rendre la page plus sophistiquée en caractérisant l'ascenseur vertical et l'ascenseur horizontal.

Attention aux couleurs

Évitez les couleurs qui ne permettent plus de distinguer la glissière de son rail ou qui font prendre le rail pour la glissière. En règle générale, il vaut mieux réserver la couleur plus foncée pour la glissière.

Tab. 4.4 : Les caractéristiques de l'ascenseur de page

Code	Désignation
`scrollbar-darkshadow-color`	L'apparence des ombres de la glissière
`scrollbar-3dlight-color`	Couleur pour les effets de relief
`scrollbar-arrow-color`	Couleur pour le pointeur de défilement
`scrollbar-base-color`	Couleur de base de la glissière
`scrollbar-face-color`	Couleur pour la surface
`scrollbar-highlight-color`	Couleur pour le bord haut et le bord gauche
`scrollbar-shadow-color`	Couleur pour le bord droit et le bord du bas
`scrollbar-track-color`	Couleur pour le rail de défilement

Prenons un exemple :

(voir Figure 4.18)

```
scrollbar-base-color: #D19C8C;
scrollbar-highlight-color: #ffffff;
scrollbar-shadow-color: #ffffff;
scrollbar-3dlight-color: #cc3333;
scrollbar-arrow-color: #cc3333;
scrollbar-track-color: #FFECCB;
scrollbar-darkshadow-color: #cc3333;
```

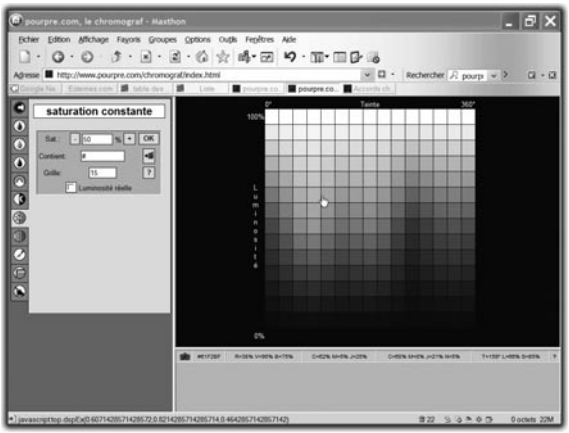
▲ Figure 4.18 : *Ascenseur coloré*

Utiliser des couleurs déjà présentes

Une solution pour éviter de faire de votre page un sapin de Noël ou de la maquiller comme un camion volé est d'utiliser les couleurs dominantes en les répétant ou en choisissant des couleurs complémentaires.

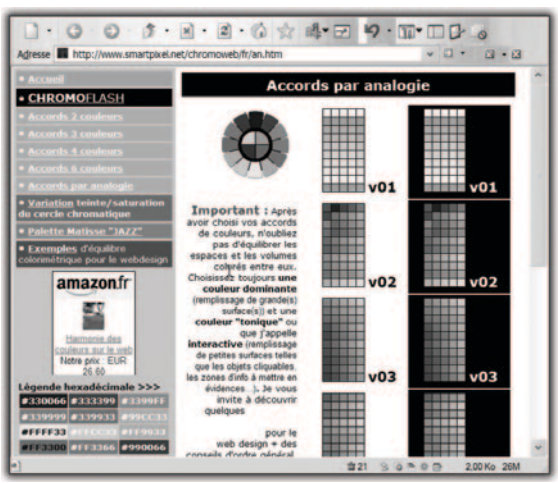
◄ Figure 4.19 : *La chemise rouge*

Deux sites indispensables pour ne pas s'emmêler les pinceaux :



▲ Figure 4.20 : *Chromographe*

`www.pourpre.com/chromograf/`



▲ Figure 4.21 : *SmartPixel*

`www.smartpixel.net/chromoweb/`

4.6 Relooker des formulaires

Les formulaires gris ou revêtus de l'uniforme Windows ou Macintosh sont peu créatifs et caractérisent mal votre site. Repeignez vos formulaires des couleurs qui expriment une personnalité. Les inventeurs des feuilles de style et des navigateurs ont pensé à vous. Ils proposent des propriétés diverses pour agrémenter les formulaires et les rendre moins administratifs et rébarbatifs.

Soumettre un formulaire

Le bouton **Submit** doit donner envie à l'usager de cliquer dessus. Pour ce bouton de formulaire, utilisez le protocole de classe en ajoutant le nom du style précédé d'un point dans la feuille de style :

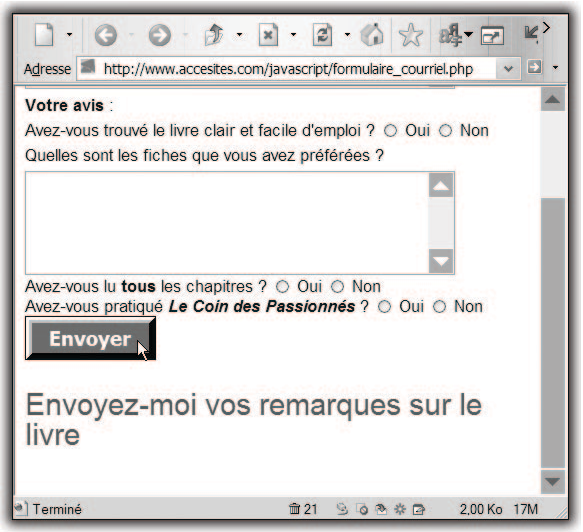
(voir Figure 4.22)

```
admin_bouton {
font-family: Tahoma, Verdana, Arial;
font-size: 10pt;
font-style: normal;
line-height: normal;
font-weight: bold;
background-color: #BC4948;
color: #FFFFFF;
}
```

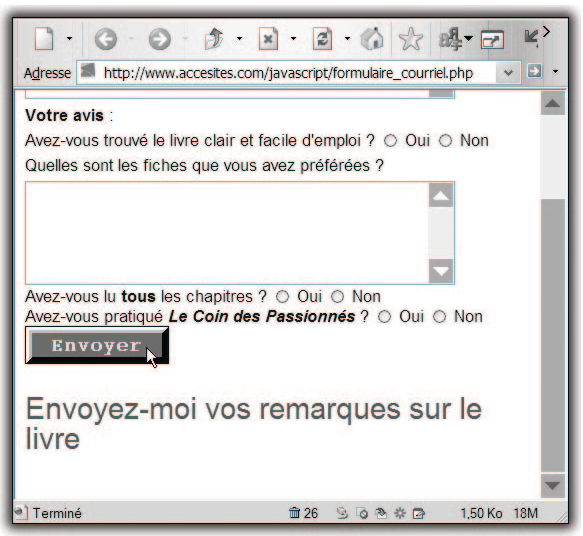
Puis dans la balise `submit`, insérez le code comme ceci :

(voir Figure 4.23)

```
<input type="submit"
   value="choisissez" class="admin_bouton">
```



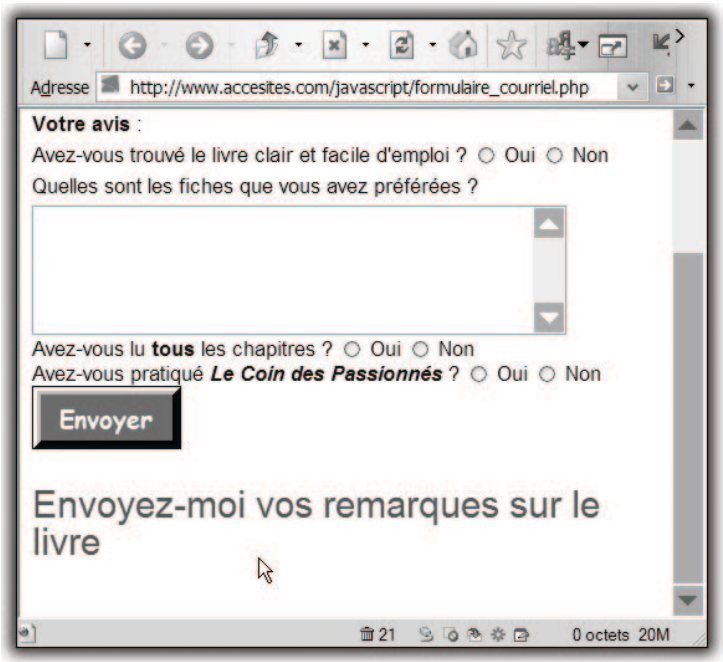
▲ Figure 4.22 : *Bouton de soumission coloré*



▲ Figure 4.23 : *Bouton de soumission avec des caractères jaunes*

Comme vous le constatez, vous êtes à même de contrôler :

- le type du caractère `font-family` ;
- la taille du caractère `font-size` ;
- le style du caractère `font-style` ;
- la hauteur de ligne pour le positionner dans le bouton `line-height` ;
- la graisse du caractère `font-weight` ;
- la couleur de fond `background-color` ;
- la couleur du caractère `color`.



▲ Figure 4.24 : *Bouton de soumission*

Vous pouvez ajouter d'autres propriétés. Si les propriétés sont absentes, elles affichent des valeurs par défaut.

Un peu de couleur sur un menu déroulant

Un menu déroulant avec ce code retrouve des couleurs et des polices dignes de ce nom. Optez pour la classe `couleur` de la balise `select`.

```
select.couleur{
font-family: Tahoma, Verdana, Arial;
font-size: 10pt;
font-style: normal;
line-height: normal;
font-weight: bold;
background-color: #607BB2;
color: #FFFFFF;
}
```

Comme vous le constatez, ce sont les mêmes propriétés que celles du bouton `submit` vu précédemment. Vous pouvez, comme pour le bouton `submit`, insérer des propriétés qui ajoute une bordure en relief :

▲ Figure 4.25 : *Menu coloré*

```
border-top-width: thick;
border-right-width: thick;
border-bottom-width: thick;
border-left-width: thick;
border-top-style: outset;
border-right-style: outset;
border-bottom-style: outset;
border-left-style: outset;
border-top-color: #FFFFFF;
border-right-color: #000066;
border-bottom-color: #000066;
border-left-color: #FFFFFF;
```

Les propriétés énumérées ici concernent la couleur (`color`), l'épaisseur (`width`) et le style (`outset`). Les couleurs reçoivent les valeurs de code hexadécimales. L'épaisseur peut être `thick` (épais), `thin` (fin) ou `medium`. Vous pouvez également définir une valeur d'épaisseur de votre choix, comme `1px` (1 pixel).

Si quatre valeurs sont attribuées à la propriété `border-width`, séparées par un espace, elles se réfèrent respectivement aux côtés haut, droit, bas et gauche. La priorité va de gauche à droite. Ainsi, s'il n'y a que trois valeurs, la plus à droite, en l'occurrence la valeur pour la bordure gauche, sera pourvue par défaut.

```
border-width: 1px thin medium thick;
```

Les styles de bordure permettent d'avoir un effet d'enfoncement (`inset`), de relief (`outset`) ou de ligne discontinue (`dashed`).

Les listes

Les listes sont de deux styles :

- à puces avec la balise `ul` ;
- ou ordonnées avec la balise `ol`.

Comme attribut, vous pouvez, comme vu précédemment, mettre une image à la place de chaque puce ou adapter des styles.

Tab. 4.5 : Différentes valeurs de la propriété list-style-type

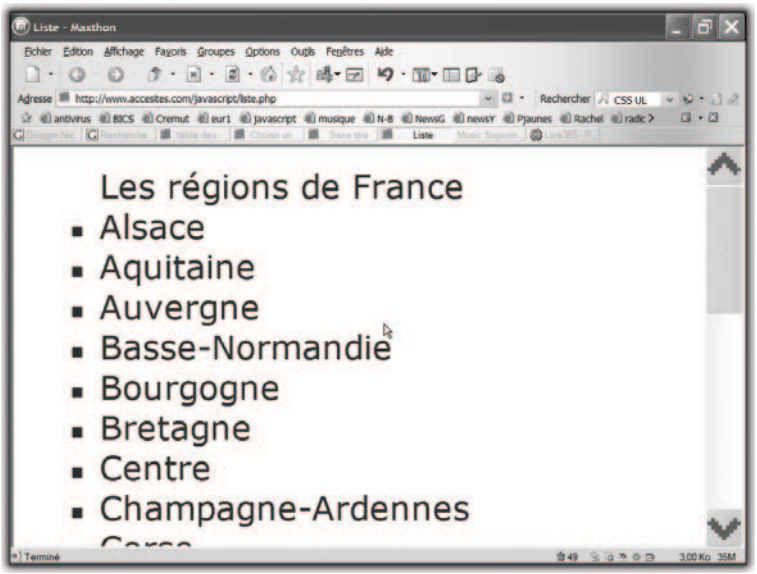
Attribut	Type de marquage	Type de liste
decimal	Numérotation 1., 2., 3., 4., etc.	ol
lower-roman	Numérotation i., ii., iii., iv., etc.	ol
upper-roman	Numérotation I., II., III., IV., etc.	ol
lower-alpha	Numérotation a., b., c., d., etc.	ol
upper-alpha	Numérotation A., B., C., D., etc.	ol
lower-latin	Numérotation a., b., c., d., etc.	ol
upper-latin	Numérotation A., B., C., D., etc.	ol
disc	Rond plein	ul
circle	Rond vide	ul
square	Puce rectangulaire	ul
none	Rien	ul ou ol

Essayez divers codes avant de trouver celui qui vous convient :

(voir Figure 4.26)

```
ul {
list-style:square;
list-style-position: outside;
color: #960000;
font-family: verdana;
font-size: 14pt;
}
```

```
ul {
list-style-image:url(puce.gif);
font-family: verdana;
font-size: 9pt;
}
```



▲ Figure 4.26 : *Liste à puces*

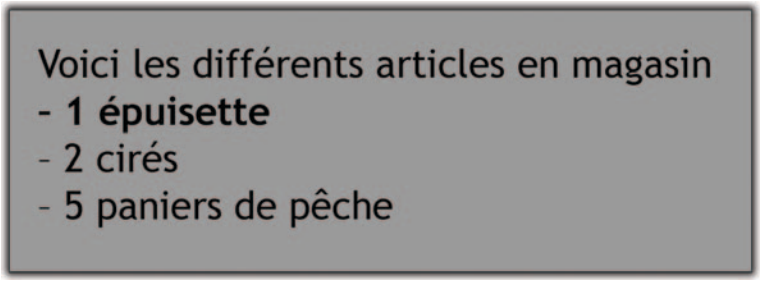
Les pseudo-classes

First-child

La pseudo-classe `first-child` permet d'agir sur le premier enfant d'un sélecteur donné. Imaginons que, dans un site, le premier paragraphe qui suit le titre des articles serve de chapeau. Au lieu de créer une classe chapeau, CSS2 permet de distinguer le premier enfant d'un sélecteur donné.

```
p:first-child {
font-weight: bold;
font-style: italic;
}
```

Le premier paragraphe de la page deviendra un chapeau en gras et en italique sans qu'il soit besoin de le préciser. Le style sera appliqué par défaut. Par contre, pour une feuille de style appliquée à tout un site, il faudra créer une classe à destination des pages sans chapeau.



▲ Figure 4.27 : *Premier enfant en gras*

```
li:first-child {
font-weight: bold;
}
```

Le premier élément de la liste est en gras. Internet Explorer ne reconnaît pas ce pseudo-élément.

Les ancres

Les liens dans la plupart des navigateurs sont marqués par une couleur par défaut selon qu'ils ont été visités ou non. Les pseudo-classes `:link` et `:visited` requalifient cette propriété. `:link` pointe vers les liens vierges et `:visited` vers les liens visités.

```
a:link {
color: #458;
}
a:visited {
color: #A79;
}
```

Il est aussi possible d'écrire le code de cette façon :

```
:link {
color: #458;
}
:visited {
color: #A79;
}
```

Les pseudo-classes `:link` et `:visited` s'appliquent uniquement aux liens.

Les pseudo-classes dynamiques

Les pseudo-classes dynamiques décrivent un état provoqué par un événement. Elles sont au nombre de trois : `:hover`, `:active` et `:focus`.

`:hover` décrit les propriétés d'un objet quand la souris le survole. Cet objet peut être un élément de liste, un lien.

▲ Figure 4.28 : *Liens et survol*

Hover sur Internet Explorer

`:hover` s'applique sur Mozilla pour un élément de liste, mais pas sur Internet Explorer.

`:active` s'applique au moment où l'internaute clique sur un lien et avant qu'il ne relâche le bouton.

`:focus` s'applique dès que l'internaute utilise son clavier, par exemple pour entrer une tabulation.

Déclarer les effets dans l'ordre

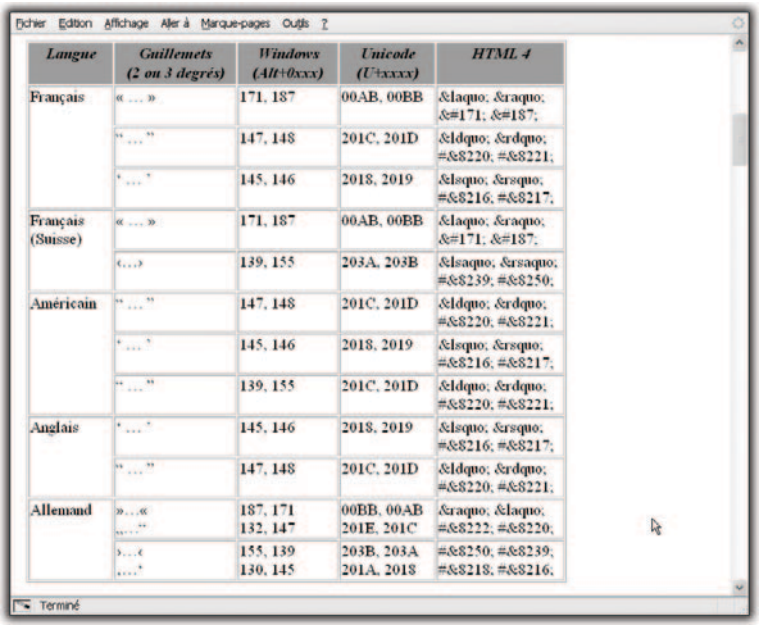
Pour que les effets s'affichent sans s'annuler, il est nécessaire de les déclarer dans un ordre précis : `:link`, `:visited`, `:hover`, `:active`.

La pseudo-classe :lang

La pseudo-classe `:lang` permet de couvrir les besoins liés à la typographie et à la syntaxe de chaque langue.

```
:lang(fr-CH) > Q { quotes: '« ' ' »' '<' '>'} /*4 arguments*/
```

Les deux premières données concernent les guillemets et les deux dernières désignent les guillemets à l'intérieur d'autres guillemets. Le code se rpésente avec quatre arguments ou les deux premiers seulement.. En français de Suisse on utilise un double chevron – et un simple chevron à l'intérieur des guillemets. Pour connaître les caractères de guillemets, allez sur `www.mus.ulaval.ca/roberge/gdrm/08-codes.htm`.



Langue	Guillemets (2 ou 3 degrés)	Windows (Alt+0xxx)	Unicode (U+xxxx)	HTML 4
Français	« … »	171, 187	00AB, 00BB	« » « »
	" … "	147, 148	201C, 201D	“ ” #&8220; #&8221;
	' … '	145, 146	2018, 2019	‘ ’ #&8216; #&8217;
Français (Suisse)	« … »	171, 187	00AB, 00BB	« » « »
	‹…›	139, 155	203A, 203B	‹ › #&8239; #&8250;
Américain	" … "	147, 148	201C, 201D	“ ” #&8220; #&8221;
	' … '	145, 146	2018, 2019	‘ ’ #&8216; #&8217;
	" … "	139, 155	201C, 201D	“ ” #&8220; #&8221;
Anglais	' … '	145, 146	2018, 2019	‘ ’ #&8216; #&8217;
	" … "	147, 148	201C, 201D	“ ” #&8220; #&8221;
Allemand	»…«	187, 171	00BB, 00AB	» «
	„…“	132, 147	201E, 201C	#&8222; #&8220;
	›…‹	155, 139	203B, 203A	#&8250; #&8239;
	‚…‘	130, 145	201A, 2018	#&8218; #&8216;

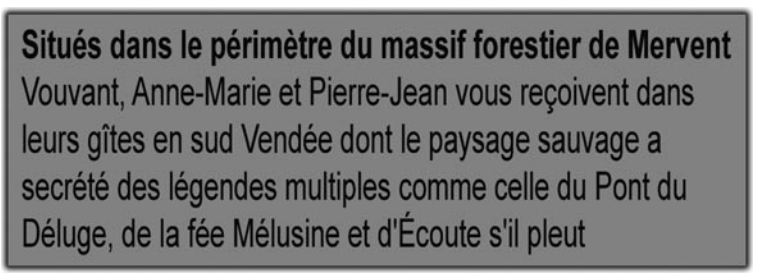
▲ Figure 4.29 : *Langues et caractères*

▲ Figure 4.30 : *Liens et langues*

Les pseudo-éléments

Le pseudo-élément :first-line

Le pseudo-élément `:first-line` s'applique à la première ligne affichée dans un paragraphe. Cette ligne est celle qui s'affiche vraiment dans le navigateur et sera donc différente selon le navigateur, la largeur affichée, la taille de la police, l'interlettrage, etc. Ce pseudo-élément ne correspond à aucune balise HTML. Pour des articles où l'auteur prévoit un chapeau sans créer une nouvelle boîte spécifique, il est possible d'utiliser ce pseudo-élément pour créer une première ligne en gras. La longueur de la première ligne dépend de l'affichage du navigateur, de la taille de la police, etc.



▲ Figure 4.31 : *Première ligne en gras*

Pour la désigner précisément, il suffit de mettre un passage à la ligne juste à la fin du chapeau avec un `<br />` et d'insérer dans la feuille de style :

```
p:first-line { text-weight: bold }
```

Dans ce pseudo-élément, il est possible de mettre toutes les propriétés de `font`, d'arrière-plan et toutes les propriétés concernant l'espacement de la ligne, l'interlettrage, etc. Ce procédé risque d'être envahissant, vu qu'il s'applique à tous les paragraphes.

Le pseudo-élément :first-letter

Cet élément spécifie les caractéristiques des lettrines. Il s'agit des premières lettres d'un paragraphe d'un corps supérieur à ce dernier et qui se tiennent sur deux, parfois trois lignes. En typographie française, les lettres suivantes sont en petites capitales. Si la lettrine commence une citation, les guillemets ouvrants viennent dans le corps du paragraphe et non dans celui de la lettrine. L'utilité de la lettrine dans une page est l'ornementation ; elle permet en outre de désigner le commencement du texte.

Sur le Web, la création d'une lettrine est un casse-tête.

```
p:first-letter { font-weight: bold; text-transform: uppercase;
font-size: 200%}
```

La première lettre des paragraphes est plus grande, plus grosse et en capitales, mais elle ne descend pas sur deux lignes, comme le fait une lettrine.

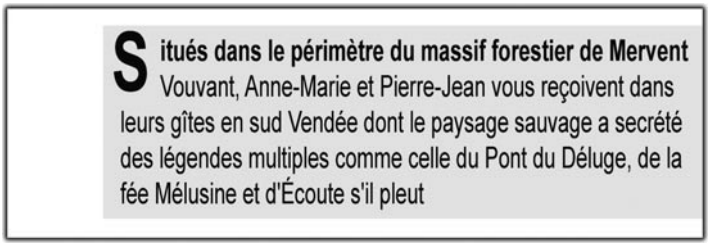
La solution est donc de mettre la lettre dans une boîte flottante.

```
.lettrine {
float: left;
font-weight: bold;
text-transform: uppercase;
font-size: 210%;
margin-right: 10px;
}
```

Il reste à mettre les lettres restantes du mot en petites capitales.

```
font-variant: small-caps;
```

Le résultat est peu harmonieux et approximatif pour beaucoup d'opérations. Si vous voulez absolument mettre des lettrines, insérez plutôt des images dans une boîte flottante.



▲ Figure 4.32 : *Lettrine*

Les pseudo-éléments :before et :after

Les pseudo-éléments `:before` et `:after` servent à insérer un objet avant ou après le contenu d'un élément. L'objet peut être un mot, une image, etc. mais pas du code HTML.

Voici des pseudo-éléments très pratiques pour créer une FAQ (foire aux questions). Ces pages sont faites de questions-réponses et ici la question est surlignée en jaune avec des caractères bleus. Le problème est que Internet Explorer n'accepte pas la génération de contenu avec la propriété `content`.

```
p.QR
{
  background-color: FFEDA6;
  color: 003D6C;
  font-family: Arial;
}
p.QR::before
{
  content: "Question: ";
  font-family: Arial;
  font-size: 1.1em;
}
p.QR::after
{
  content: " ?";
  font-family: Arial;
  font-size: 1.1em;
}
```

Vu le problème avec Internet Explorer, il est peut-être plus judicieux de placer le point d'interrogation avec son espace à la main.

▲ Figure 4.33 : *FAQ www.jardinauxpapillons.com*

Les filtres

Plusieurs filtres sont à la disposition du développeur pour créer des effets sur des images ou des textes. Ils sont reconnus uniquement par Internet Explorer 4 et les versions supérieures. Ils ne peuvent donc être utilisés que s'ils ne sont pas essentiels. Ils peuvent sembler des gadgets mais certains sont intéressants pour donner un style aux images d'un site. La propriété ne fonctionne que dans une balise en ligne. Sinon, le conseil est de créer ces effets directement sur l'image avec un logiciel graphique.

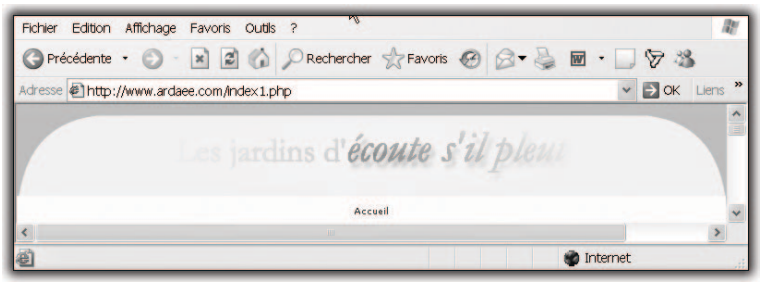
alpha

Le filtre `alpha` crée dans l'image une transparence. Le paramètre `opacity`, qui contrôle l'opacité, est obligatoire : 0 correspond à la transparence maximale, et 100 à l'opacité maximale.

`style` permet de choisir les types de dégradés : 0 pour uniforme, 1 pour linéaire et 2 pour radial.

Avec `style=1`, pour intégrer un dégradé, il faut nécessairement indiquer une valeur de départ (`startx` et `starty`) et de fin (`finishx` et `finishy`) en pixels ou en pourcentage.

```
<img style="filter: alpha(apacity=50, finishopacity=0, style=2,
 startxX=20, starty=40, finishx=0, finishy=0)"
src="images/titre_ecoute_vrai_1.jpg">
```



▲ Figure 4.34 : *alpha*

shadow

Ce filtre comporte deux paramètres : la couleur en code hexadécimal et la direction en degrés. L'ombre, contrairement à l'ombre portée de `dropshadow`, est diffuse.

```
filter: shadow (color=#CACACA, direction=45);
```

dropshadow

Ce filtre crée une ombre portée, comme si l'image ou le texte se tenaient au-dessus d'une surface. Cet effet donne du relief.

Il accepte quatre paramètres :

- la couleur de l'ombre portée ;
- le décalage horizontal et le décalage vertical (`offX` et `offY`) ;
- la position, dont la valeur est un booléen : 1 pour une ombre à l'extérieur, 0 pour une ombre à l'intérieur de l'élément.

```
filter: dropshadow (color=##83838C, offX=3, offY=3, positive=1);
```

glow

Ce filtre ajoute un halo de couleur autour de l'image ou du texte. Il accepte deux paramètres : la couleur et la densité.

```
filter: glow (color= #D7B5CF, strength=5);
```

chroma

Ce filtre permet de définir, dans une image GIF, une couleur comme transparente.

```
<img src="image.gif" style="filter:chroma(color=#006699)"
width="133" height="53" alt="image" border="0">
```

Selon le type de média

Avec CSS2 apparaît la prise en compte du format du média visé.

Tab. 4.6 : Les différents types de médias

Type	Application
`media="all"`	Tous les types de médias.
`media="aural"`	Synthèse vocale assistée par ordinateur.
`media="braille"`	Ce type de média est paramétré pour les aveugles. Le texte est traduit sur une surface modifiée en braille, ce qui permet aux aveugles de lire le texte.
`media="embossed"`	Le texte est pressé en relief, sur une imprimante braille, sous forme de structures de surface pouvant être reconnues au toucher sur du papier ou sur un matériau comparable, à destination des aveugles.

Tab. 4.6 : Les différents types de médias

Type	Application
`media="handheld"`	Ce format s'applique à des assistants numériques, des téléphones portables et d'autres agendas électroniques. Le format est prévu pour des petits écrans.
`media="print"`	Ce format s'applique à l'imprimante quand l'utilisateur désire imprimer.
`media="projection"`	Ce format concerne les rétroprojecteurs et d'autres projecteurs vidéo.
`media="screen"`	C'est le format par défaut. Il s'applique aux écrans d'ordinateur.
`media="tty"`	Ce média concerne les sorties texte avec une largeur invariable, comme les télex. Il est aussi intéressant pour les navigateurs texte de type Lynx.
`media="tv"`	Ce format est celui de l'audiovisuel, comme la télévision, où le son et l'image se combinent. La qualité visuelle n'a pas besoin d'être parfaite.

Pour les navigateurs qui n'implémentent pas CSS2, on emploie généralement la règle `@import`, qui permet d'importer les feuilles de style. Elle n'est pas reconnue par les anciens navigateurs.

```
<html>
<head>
<title>Titre du fichier</title>
<style type="text/css">
<!--
@import url(impression.css) print, embossed;
@import url(portable.css) handheld;
@import url(normal.css) screen;
-->
</style>
</head>
<body>
</body>
</html>
```

La règle @import

La règle `@import` s'insère dans une feuille de style, qu'elle soit interne ou externe.

Pour imprimer, il est préférable de gommer les couleurs et d'avoir le meilleur contraste possible entre le texte et le fond de page. Les menus ne sont pas utiles. En revanche, les liens de référence qui éclairent le propos par des pages annexes, doivent être explicites et complets. La feuille de style `print` doit tenir compte de tout cela. Le code qui suit est tiré du site `www.ardaee.com` et permet de ne pas afficher les images, d'avoir des caractères noirs sur un fond blanc, de décliner les liens en entier et en clair, de ne pas imprimer les menus de navigation, d'afficher une marge sur des navigateurs compatibles CSS2. Internet Explorer imprimera bien la page mais sans les liens en clair. De ce fait, il est utile d'insérer le nom de domaine dans le pied de page.

```
body {
color: #000;
background: #fff none;
font-family: arial, sans-serif;
font-size: 10pt;
line-height: 12pt;
}
#navidroite, #navigauche {
display: none;
}
img, #titre img {
display: none;
}

#corps {
   width: auto;
   margin: 0 5%;
   padding: 0;
   border: 0;
   float: none !important;
   color: black;
   background: transparent none;
   }
.pied {
   margin: 3% 10%;
```

```
   }
a {
   color: gray;
   background: transparent;
   font-weight: bold;
   text-decoration: none;
   }

a:link:after {
   content: "(" attr(href) ")";
   font-size: 90%;
   }
a[href]:after {
   content: " (http://www.ardaee.com/" attr(href) ") ";
   }
a[href^="http://"]:after {
   content: " (" attr(href) ") ";
   }
a[href^="#"]:after {
   content: "";
   }
```

Les liens sont mis en gris et les liens relatifs sont transformés en absolus. Comme il n'existe pas de recherche d'un "non-motif", c'est-à-dire de mots qui ne contiennent pas le motif, l'astuce consiste à ajouter le lien directeur du site à tous les liens puis de l'enlever sur les liens externes. Le dernier sélecteur enlève les ancres qui envoient vers un point de la page elle-même.

◄ Figure 4.35 : *Prête à imprimer*

Bidouillages et défaillances autour d'Internet Explorer

Cependant, la venue d'Internet Explorer 7, qui ajoute encore de la complexité sans apporter de la conformité, va compliquer davantage le développement des sites. Il faut espérer que ce navigateur perde des parts de marché au profit de Firefox ou d'autres navigateurs compatibles pour simplifier le développement des sites.

Le problème des bidouillages de la feuille de style est que celle-ci n'est plus conforme et peut amener des bogues avec les navigateurs du futur.

Mac ou PC

Les versions Mac et PC d'Internet Explorer ont été créées par deux équipes indépendantes, qui ont donc produit deux logiciels différents.

Le sélecteur * html (révélé à Internet Explorer PC seulement)

Ce sélecteur représente la balise `html` comme descendant d'un élément quelconque, ce qui est une sélection impossible car la balise `html` est l'élément racine des pages HTML. Un navigateur normal ignore ce sélecteur car il ne correspond à rien. Internet Explorer lui en tient compte.

```
div#corps {
position: absolute;
max-width: 600px;
margin: 0 250px;
padding: 0 20px 20px 20px;
background-color: #E2EA8A;
}
* html div#corps {
width: 600px;
}
```

Le problème avec Internet Explorer est que, si la propriété `width` n'est pas présente, la largeur n'a plus de limites. Ici le sélecteur `* html div#corps` n'est reconnu que par lui.

Le sélecteur enfant, adjacent ou le sélecteur d'attribut (caché à Internet Explorer PC seulement)

Pour cacher certaines règles à Internet Explorer PC, il suffit d'utiliser le sélecteur enfant qu'il ne reconnaît pas.

```
html>body div#corps {
max-width: 600px;
}
```

Ici, il ne reconnaît pas la propriété `max-width`, mais elle lui est cachée par la présence du caractère >, qui désigne un sélecteur enfant.

```
td+td+td:after {
content:url(acheter.png);
}
```

Ici, les acheteurs potentiels qui utilisent Internet Explorer ne pourront rien acheter car ils ne verront pas l'image qui les renseigne. En effet, le sélecteur est adjacent et le pseudo-élément `:after` n'est pas reconnu, ni `content` d'ailleurs.

Les sélecteurs d'attribut ne sont pas reconnus par Internet Explorer :

```
a[href|="http"] {
color:#261;
}
```

Le commentaire (caché à Internet Explorer Mac seulement)

Internet Explorer Mac voit l'antislash comme un signe d'échappement, y compris dans un commentaire. Par conséquent, dans ce qui suit, il ne voit pas la fin du commentaire de la première ligne et il prend en compte la deuxième ligne de code.

```
.nav {width:200px}
/* IE MAC ne voit pas la fin de commentaire à cause de
l'antislash \*/
     .nav {width:150px}
 /* IE MAC voit un commentaire sur 3 lignes, cette ligne
 incluse */
```

nav a une largeur de 150 px, sauf pour Internet Explorer Mac qui affiche une largeur de 200 px.

Le commentaire (révélé à Internet Explorer Mac seulement)

Internet Explorer pour Mac ne voit pas l'astérisque masqué par un antislash et qui ferme le commentaire pour tous les autres navigateurs.

```
     .nav {width:200px}
 /*IE Mac ne voit pas l'astérisque qui suit l'antislash \*//*/
     .nav {width:250px}
 /**/
```

nav a une largeur de 200 px, sauf pour Internet Explorer Mac qui voit une largeur de 250 px.

Le blanc souligné (révélé à Internet Explorer PC seulement)

Internet Explorer PC ignore le blanc souligné apposé devant une propriété. Les autres navigateurs ne le prennent pas en compte et le voient comme une erreur de code.

```
     .nav {_width:250px;width:200px}
```

nav a une largeur de 200 px, sauf pour Internet Explorer PC qui voit une largeur de 250 px.

min-height et min-width (cachés à Internet Explorer PC et Mac)

min-height et min-width, max-height et max-width sont des propriétés indispensables pour des conteneurs dont on ne connaît pas la taille du contenu et surtout dans un contexte où l'architecture de la page est

extensible selon la définition de l'écran de l'internaute. Or Internet Explorer les ignore, ce qui pose des problèmes pour la mise en page avec des colonnes sous forme de boîtes.

Une solution consiste à utiliser une image de fond pour avoir une hauteur ou une largeur minimale. Il est aussi intéressant de mettre un `width` ou un `height` dans un sélecteur qui est vu uniquement par Internet Explorer. Reprenons le premier exemple de cette section.

```
div#corps {
position: absolute;
max-width: 600px;
margin: 0 250px;
padding: 0 20px 20px 20px;
background-color: #E2EA8A;
}
* html div#corps {
width: 600px;
}
```

Pour remédier au fait que Internet Explorer ignore la propriété `min-height`, la solution consiste en un ensemble de deux conteneurs, l'un dans l'autre. Le conteneur extérieur a un `padding` équivalent à cette hauteur minimale requise et le conteneur intérieur est doté d'une marge négative équivalente.

```
<style type="text/css">
  #out-min-height {
  padding-top:200px;
/* voici le min-height (200px) */
  min-height: 1px;
 /* pour Opéra */
  }
  #in-min-height {
  margin-top: -200px
/* pour compenser le padding-top */
  }
</style>
<body>
<div id="out-min-height">
<div id="in-min-height">
Votre contenu ici, long ou court à votre guise.
</div>
</div>
</body>
```

margin et padding

Les marges internes et externes sont gérées différemment par les navigateurs. Chacun fait sa "cuisine" pour définir les marges des éléments. C'est pourquoi, dès l'entrée, il est préférable de remettre tout à 0.

```
body {
margin: 0;
padding: 0;
}
```

Cette portion de code aide à positionner des éléments dans une page de manière plus précise en mettant tous les navigateurs sur le même réglage. Sinon les marges par défaut des différents navigateurs vont fausser l'affichage.

Internet Explorer 7

Internet Explorer 7 est un projet dirigé par Dean Edwards, qui consiste en une bibliothèque JavaScript compensant les manques d'Internet Explorer. Ces caractéristiques sont les suivantes :

- Il répond aux sélecteurs CSS suivants :
  - parent > enfant ;
  - frère + adjacent ;
  - frère ~ adjacent ;
  - `[attr]`, `[attr="valeur"]`, `[attr~="valeur"]`…
  - Classes multiples (réparent certains bogues).
  - `:hover`, `:active`, `:focus` (pour tous les éléments).
  - `:before`, `:after`, `content:`.
  - `:lang()`.
- Il fonctionne avec des documents HTML ou XML indifféremment.
- Il applique les feuilles de style importées.
- Il préserve la cascade dans la feuille de style.
- Il ne modifie pas la structure du document.
- Il ne fait pas appel sans cesse à la structure DOM du document.
- Il utilise du pur CSS pour suppléer aux règles CSS.

- Il accepte le modèle de boîte du W3C.
- Il accepte la position `fixed` (exempte de scintillement).
- Il accepte `overflow:visible`.
- Il accepte `min/max-width/height`.
- Il accepte la transparence alpha des PNG.
- Script poids léger (22 K).
- Il est entièrement modulaire.
- Il fonctionne pour Microsoft Internet Explorer 5+ (Windows seulement).

Comme cette application en JavaScript est formatée en modules, le développeur utilise les modules dont il a besoin sans être obligé de prendre l'application entière. La meilleure façon de faire est de construire le site normalement, puis de tester dans divers navigateurs et de remédier aux problèmes avec les bidouillages proposés. Enfin, si certaines erreurs sont rédhibitoires, utilisez Internet Explorer 7 pour les gommer. Il faut quand même prendre en compte le fait que certains internautes ont désactivé la fonction JavaScript sur leur navigateur.

Ce genre d'application est un gain de temps et permet d'utiliser par exemple les pseudo-éléments `:before` et `:after` pour créer des cadres, ou du moins de placer un haut pour les coiffer et un bas pour les chausser.

```
#cadre:before {
content:url(haut.png);
}
#cadre:after {
content:url(bas.png);
}
#cadre {
background-color:#b9b9b9;
border: 1px solid #555;
width:200px;
}
```

Ce genre de code est infaisable habituellement car Internet Explorer n'accepte pas ces pseudo-éléments.

On trouve Internet Explorer 7 sur le site de Dean Edwards (`http://dean.edwards.name/IE7/`).

hr

<hr /> est la balise utilisée pour figurer une ligne de séparation. Elle est peu employée. Mais avec les CSS, elle reprend du service.

Pour avoir un trait de séparation simple avec une CSS, il faut un code assez simple.

```
hr {
  border: 0;
  width: 80%;
}
```

Pour créer un trait plus épais en bleu, il faut caractériser le fond et le contour de la même couleur.

```
hr {
color: #00F;
background-color: #00F;
height: 5px;
}
```

Ici, il faut utiliser, à la fois, les propriétés `color` et `background-color`. La première propriété est insérée pour Internet Explorer tandis que la deuxième est présente pour les autres navigateurs.

Il est intéressant d'utiliser une image comme séparateur pour afficher un motif grâce à une astuce.

```
<div class="hr"><hr /></div>
```

Ce code XHTML est nourri par la feuille de style :

```
div.hr {
height: 15px;
background: transparent url(motif.gif)
no-repeat scroll center;
}
div.hr hr {
  display: none;
}
```

Pour Internet Explorer, <hr /> est un élément en ligne d'une hauteur de 7 pixels par défaut avec des `padding` haut et bas figés à 7 pixels. Avec un

display: block et une marge négative de 7 pixels, le trait est affiché correctement sur Internet Explorer.

```
<style type="text/css">
 hr {
 display:block;
 height: 1px;
 margin: 0;
 _margin: -7px 0;
 padding: 0;
 color: #03A;
 background-color: #03A;
 border: 0;
 }
</style>
<body>
     <div>Premier contenu</div>
     <hr />
     <div>Second contenu</div>
</body>
```

Le blanc souligné devant margin rend cette propriété invisible à tout autre navigateur que Internet Explorer.

Ici les deux contenus sont séparés d'une ligne bleue de 1 pixel d'épaisseur sans aucune marge. Une fois les marges à 0, tout ce que le développeur ajoute comme marge dans les div ou dans le hr – en plus des 7 pixels de marge négative – .est comptabilisé, par exemple margin: 6px appose une marge d'1 px.

4.7 Conclusion

Après avoir abordé les différents aspects du code, il s'agit de le vérifier sur le terrain et de compléter sa connaissance au contact de la "réalité" dans le prochain chapitre.

5

Études de terrain

Une étude de terrain pour récolter des connaissances en CSS demande d'avoir Firefox et son extension Web Developer. Cette extension permet de découvrir ce qui se cache derrière un beau site.

Les peintres vont au Louvre, les écrivains lisent des livres puis écrivent "à la mode de", les développeurs de sites web vont à la pêche avec Firefox et Web Developer, non pour plagier, mais pour emprunter des éléments intéressants afin d'enrichir leur expression personnelle et surtout de savoir comment a été fabriqué un site.

La feuille de style est parfois illisible sans le code source, qui lui-même est illisible car trop touffu, trop long, trop complexe. La solution est d'utiliser l'éditeur de feuille de style de Web Developer et de constater les changements opérés sur la page du site, à la suite d'une modification de couleur entre autres.

Cette galerie de sites met en avant de bonnes pratiques ou du code particulièrement intéressant. Parfois l'ergonomie ou l'aspect visuel sont remarquables. Ce chapitre permet de vérifier in situ les différents aspects d'un site web.

5.1 Quelques sites

Les quelques sites présentés ont été choisis dans des domaines différents, pour leur charte graphique harmonieuse et leur approche originale du code ou d'une partie du code CSS.

Un site communicant

Le premier site communique beaucoup et bien autour de ses choix de code CSS et de leurs conséquences (peu importe le code relatif aux menus dynamiques). L'auteur a une volonté de partager ses trouvailles, de bénéficier de l'écho des webmasters qui l'utilisent et ainsi de l'améliorer. C'est exactement la philosophie des applications distribuées sous des licences "open source".

L'auteur désigne les éléments invariables du code et les éléments qui peuvent être modifiés sans interférer avec le fonctionnalité du menu. C'est une vraie documentation, qui fournit des indications concrètes sur la page, ainsi que sur des classes ou des identifiants.

▲ Figure 5.1 : *Le site iubito*

Le script a été trouvé sur le site `http://iubito.free.fr`. Il est proposé sur le site `www.openweb.eu.org`, la référence en matière de CSS et d'accessibilité.

▲ Figure 5.2 : *Openweb.eu.org*

```
.ssmenu ul, .ssmenu li {
/* Ne pas trop toucher.
   Les items (les <li>) des sous-menus, notamment pour
   enlever les puces des listes */
   padding:0;
   margin:0;
   list-style-type:none;
}

.menu a, .ssmenu a, .menu a:visited, .ssmenu a:visited {
/* Les liens qui ne sont pas sous le curseur. */
   color:white;
   padding:2px 1px;
   text-decoration:none;
/* Ne pas modifier le block, margin et width */
   display:block;
   margin:0px;
   width:100%;
}
html>body .menu a, html>body .menu a:visited, html>body
.menu a:hover, html>body .menu a:active,
html>body .ssmenu a, html>body .ssmenu a:visited, html>body
.ssmenu a:hover, html>body .ssmenu a:active {
   width:auto;
}

.ssmenu ul:hover, .ssmenu a:hover, .ssmenu a:focus,
.menu a:hover, .menu a:focus {
/* Les liens sous le curseur.
   On change les couleurs du texte et de fond, et on
enlève le soulignement.   */
   background-color:#EFB;
   color:black;
   text-decoration:none;
}

img {
/* Si on met une image dans un lien (par exemple
un logo à gauche du lien), pas de bordure
   car c'est vilain :-p */
   border:none;
}

.ssmenu img {
/* un petit espacement entre l'image et le texte du lien
   <a href="…"><img…/>blabla</a> */
   margin-right:2px;
}
```

```
.ssmenu img.hr {
/* Encore un peu expérimental, un <hr /> entre deux items
pour faire un "séparateur"… */
   margin-top:1px;
   margin-bottom:1px;
   margin-left:0;
   margin-right:0;
   width:100%;
   display:block;
}

/*
Bref… pour résumer, ce qui est à modifier à la base, c'est :
- couleurs, police, bordure.
- l'effet spécial IE pour les ssmenus si on aime ça…
- police et couleurs des liens "non survolés" et "survolés".
*/
```

Voilà comment il faut écrire son code. Les commentaires sont dignes d'un manuel d'utilisation.

Un site appétissant

Quand le visiteur arrive sur le site marmiton.org, tout festonné de fruits et de légumes, il a déjà l'eau à la bouche en voyant les différentes recettes proposées. Le site est visible en 800×600 pixels et optimisé pour $1\,024 \times 768$ pixels. Au-dessous de la bannière principale figure une bannière de publicité, encadrée par des guirlandes de fruits et légumes qui donnent un effet de relief et cassent l'arrangement géométrique de l'ensemble. Les couleurs sont printanières et se déclinent dans les verts, les jaunes et les orangés, avec une pointe de rouge (voir Figure 5.3).

Le site se dévoile sur la droite avec le menu donnant accès à la version en anglais et un feston de fleurs et de fruits (voir Figure 5.4).

▲ Figure 5.3 : *Marmiton.org*

▲ Figure 5.4 : *Guirlandes et festons*

La navigation se fait au moyen d'onglets sis sur la bannière, qui se déroulent sous forme de menus dynamiques en CSS. La page d'accueil est structurée sous forme de boîtes autour des recettes, avec un moteur de recherche, une mise en avant d'un plat ou des propositions de menus. Le site est irrigué par une communauté de membres. Pour accéder à l'espace personnel, il faut cliquer sur l'onglet. Le formulaire d'identification apparaît.

▲ Figure 5.5 : *Menus*

Sur ce site, le développeur s'arrange astucieusement pour qu'on oublie les cadres et donner du relief à chaque boîte de rubrique. En regardant la feuille de style, on découvre la cuisine du site.

Chaque boîte est coiffée d'une image qui donne l'impression du relief, qui casse les lignes droites, tout en arrondissant les angles. Certaines de ces boîtes sont pourvues d'une deuxième image qui leur sert de socle.

◄ Figure 5.6 : *Boîte du moteur de recherche*

Le code source de la boîte de recherche dans les recettes se présente dans une table et une colonne.

```
<div id="mrm_content_main">
<table id="mrm_content_table">
   <tr valign="top">
      <td>
      <!-- colonne gauche -->
      <div id="mrm_content_column1">
      <!-- panneau de recherche -->
      <div id="mrm_panel_search">
      <div id="mrm_panel_search_2">
      <div id="mrm_panel_search_recipe">      <img />
      <div id="mrm_panel_search_recipe_info"><form>
   </form>
       </div>
       </div>
       <div id="mrm_panel_search_idea">
         <img />
      <div id="mrm_panel_search_idea_info">
      <form>…
```

```
      </form>
     </div>
     </div>
     </div>
</div>
```

La page est faite de tables imbriquées qui encadrent des boîtes. La boîte de recherche de menus est constituée de six boîtes imbriquées. Le style des colonnes précise une position relative, excepté pour la colonne 3, dont les codes ont été désactivés sous forme de commentaire.

```
#mrm_content_table{
   width: 100%;
   margin: 0px;
   padding: 0px;
}
.mrm_content_table_col1{
   width: 246px;
}
.mrm_content_table_col2{
   width: 231px;
}
.mrm_content_table_col3{
   width: 225px;
}
#mrm_content_column1{
   position: relative;
   top: 24px;
   left: 20px;
   width: 469px;
}
/*#mrm_content_column2{
}*/
#mrm_content_column3{
   position: relative;
   width: 702px;
}
```

Le principe des tables et des boîtes n'est pas orthodoxe, mais il est efficace dans tous les navigateurs. Par contre, il déroge aux principes d'accessibilité puisque les tables ne sont là que pour assurer la forme générale de la page et distribuer les différents éléments sans qu'aucune propriété comme `summary` ne le formule clairement. Le chapitre suivant passe en revue quelques règles. Cela dit, l'habillage de ces boîtes est remarquable. Pour habiller la boîte de recherche, le développeur utilise trois images qui sont insérées en tant qu'images de fond sur trois boîtes imbriquées. Trois

propriétés permettent à ces images d'être positionnées là où elles doivent l'être : `background-image:url`, `background-position` et `background-repeat`.

```
/*************** Panel : Search ********************/

#mrm_panel_search{
   position: absolute;
   top: 0px;
   left: 0px;
   background-color: #E6E731;
   background-image:url("panel_search_top.gif");
   background-position:top;
   background-repeat: no-repeat;
   width: 227px;
   height: 258px;
   padding: 0px;
   margin: 0px;
   /*border: solid 1px #FFE16D;*/
}

#mrm_panel_search_2{
   position: absolute;
   height: 258px;
   background-image: url(panel_search_bottom.gif);
   background-repeat:no-repeat;
   background-position:bottom center;
   padding: 0px;
   margin: 0px;
}

#mrm_panel_search_recipe{
   position: relative;
   top: 17px;
   background-image:url("panel_search_leaf.jpg");
   background-repeat:no-repeat;
   background-position:bottom left;
   height: 135px;
}

#mrm_panel_search_recipe_info{
   padding-left: 20px;
   padding-right: 20px;
   padding-bottom: 25px;
   text-align: center;
}

#mrm_panel_search_idea{
   position: absolute;
```

```
   top: 156px;
}

#mrm_panel_search_idea_info{
   display: block;
   margin: 0px;
   padding-top: 5px;
   padding-left: 20px;
   padding-right: 20px;
   text-align: center;
}
/*************** End Panel : Search *******/
```

Les différents éléments sont clairement étiquetés et donnent des indications claires à la personne chargée de maintenir le site.

Le site est beau, bien conçu, pensé pour séduire le visiteur par son design et la présentation du contenu, ainsi que par le ton et les différentes aides qui guident le nouveau visiteur.

Un site design

Le site de Jérémie, `www.imerege.com`, est original par son design propre et clair. Sa charte graphique est cohérente, entre le vert et l'orange. Le jeu sur la typographie concourt à l'harmonie de l'ensemble (voir Figure 5.7).

La page d'accueil (`www.imerege.com/home.php`) est faite de boîtes qui contiennent d'autres boîtes. Les cinq boîtes principales sont `conteneurheader`, `conteneurmenu`, `conteneurcontenu`, `contenunews` et `conteneurcopyright`, toutes contenues dans un conteneur général.

```
conteneurgeneral{
   width: 770px;
   margin-right: auto;
   margin-left: auto;
   text-align: left;
}
```

La boîte **conteneurgeneral** a une largeur définie et des marges qui s'adaptent à la largeur de l'affichage du navigateur pour se centrer sur un fond noir.

La présentation est claire, avec une hiérarchie qui part du général au particulier, c'est-à-dire que la feuille de style commence avec les

sélecteurs `body`, `a`, `img`. Puis le développeur décline le conteneur général puis chaque conteneur important en partant du conteneur d'en-tête jusqu'au conteneur du copyright. Chaque partie est clairement séparée par des lignes blanches et une enseigne formée de pointillés encadrant le titre.

▲ Figure 5.7 : *imerege.com*

```
body {
   font-family: Verdana, Arial, Helvetica, sans-serif;
   font-size: 9px;
   color: #FFF;
   margin-top: 12px;
   padding: 0;
   background-color:#000;
   text-align:center;
   scrollbar-shadow-color: #294931;
```

```
   scrollbar-darkshadow-color: #294931;
   scrollbar-3dlight-color: #294931;
   scrollbar-highlight-color: #294931;
   scrollbar-face-color: #294931;
   scrollbar-track-color: #294931;
   scrollbar-arrow-color: #546656;
}
a{
   text-decoration: none;
   font-weight: bold;
   color: #FFF;
}
a:hover{
   text-decoration: underline;
}
img{
   border: 0;
}
```

Les indications concernant l'ascenseur sont vues uniquement par les navigateurs Internet Explorer et compatibles.

▲ Figure 5.8 : *Le conteneur d'en-tête*

La boîte `conteneurheader` contient tout ce qui apparaît en en-tête sur la page. Elle contient surtout l'image qui permet d'identifier le site et la boîte pour l'édito posée par-dessus.

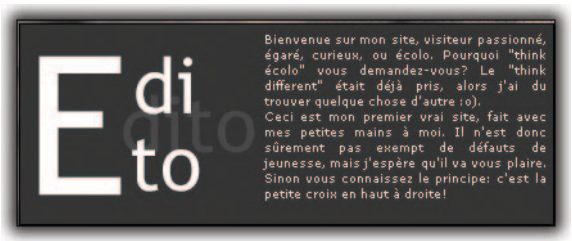
```
conteneurheader{
   height: 239px;
   background-image: url(images/header_fond_automne.jpg);
   border-top: 1px solid black;
}
```

Dans le site, chaque rubrique est annoncée remarquablement par un mot stylisé accompagné en filigrane par un mot écrit simplement. Le seul

bémol est l'accessibilité : cette étiquette est une image de fond qui ne contient pas de balise `alt` pour donner une légende à l'image.

```
<div class="conteneurgeneral">
   <div class="conteneurheader">
    <div class="edito">
         <div class="editohome"></div>
      <p class="editotxt">
           Bienvenue sur mon site…
        </p>
     </div>
   </div>
```

Le conteneur `edito` contient le conteneur `editohome`. Ce dernier ne contient rien excepté une image de fond qui est l'étiquette édito. La position est flottante à gauche. Le texte, à travers la classe `editotxt`, se colle directement à l'image. Tout est bien ordonné car après `editohome`, vient le style edito de chaque page du site, qui est le même partout. L'auteur a sans doute voulu se garder toute latitude pour modifier une boîte `edito` pour une page donnée, sinon il aurait utilisé la classe `editohome` pour chaque page.

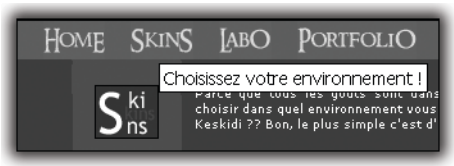


▲ Figure 5.9 : *Le conteneur edito*

```
/*----------- HEADER --------*/
.edito{
   border: 1px solid black;
   width: 405px;
   height: 145px;
   background-color: #214129;
   margin: 20px 0 0 345px;
   padding: 5px;
   text-align: justify;
}
.editohome{
```

```
   width: 185px;
   height: 145px;
   float: left;
   background-image: url(images/header_edito_automne.gif);
}
.editoskins{
   width: 185px;
   height: 145px;
   float:left;
   background-image: url(images/header_skins_automne.gif);
}
.editotxt{
   margin-top: 0;
}
```

Chaque lien possède un titre et affiche une légende au passage de la souris. Ces boîtes sont seulement des éléments d'une liste sans marqueur.



◄ Figure 5.10 : *Le menu et le titre au passage de la souris*

```
<div class="conteneurmenu">   <ul class="menu">
<li><a class="menuhome" href="home.php" title="Home"></a></li>
<li><a class="menuskins" href="skins.php"
title="Choisissez votre environnement !"></a></li>
      <li><a class="menulabo" href="labo.php"
title="Mon espace d'expérimentation"></a></li>
      <li><img src="images/menu_portfolio_automne.gif"
alt="en construction" title="en construction"
width="94px" height="30px" />
<!--<a class="menuportfolio" href="portfolio.php"
title="Mon portfolio professionnel"></a>--></li>
      <li><a class="menuwebmaster" href="webmaster.php"
title="Moi, moi et moi"></a></li>
      <li><a class="menuhumeurs" href="humeurs.php"
title="Humeurs"></a></li>
      <li><a class="menuliens" href="liens.php"
title="Ma sélection de liens"></a></li>
      <li><a class="menucontact" href="contact.php"
title="Me contacter"></a></li>
   </ul>
</div>
```

Normalement, il n'est pas possible de cliquer sur un lien qui ne contient pas de mot ou d'image. Ici en fait, le lien est formé de tout le bloc grâce à la propriété `display: block`. Le bloc est formé, non seulement par l'élément de la liste, mais aussi par l'image de fond qui lui donne sa surface. Chaque bloc a pour largeur la valeur de la propriété `width`. Cette astuce est la bienvenue dans la mesure où, sans elle, Internet Explorer ne reconnaît pas un élément de liste comme un bloc, malgré la propriété `display: block`.

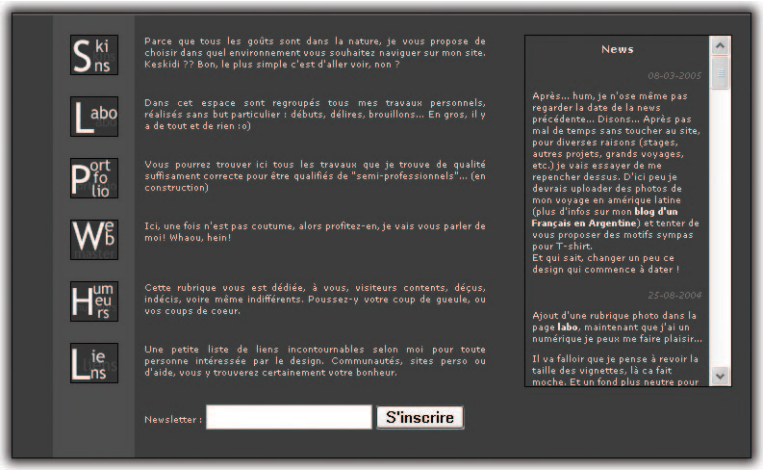
```
.conteneurmenu{
   background-color: #526552;
   height: 30px;
   border-top: 1px solid black;
}
/*---- MENU ------------*/

ul{
   margin: 0;
   padding: 0;
}
li {
 float:left;
 list-style-type: none;
 padding: 0px;
 margin: 0 0 0 20px;
 border: 0px;
}
a.menuhome{
   width: 48px;
   height: 30px;
   background-image: url(images/menu_home_automne.gif);
   display: block;
}
a.menuskins{
   width: 49px;
   height: 30px;
   background-image: url(images/menu_skins_automne.gif);
   display: block;
}
a.menulabo{
   width: 41px;
   height: 30px;
   background-image: url(images/menu_labo_automne.gif);
   display: block;
}
a.menuportfolio{
   width: 94px;
```

```
   height: 30px;
   background-image: url(images/menu_portfolio_automne.gif);
   display: block;
}
a.menuwebmaster{
   width: 104px;
   height: 30px;
   background-image: url(images/menu_webmaster_automne.gif);
   display: block;
}
a.menuhumeurs{
   width: 79px;
   height: 30px;
   background-image: url(images/menu_humeurs_automne.gif);
   display: block;
}
a.menuliens{
   width: 41px;
   height: 30px;
   background-image: url(images/menu_liens_automne.gif);
   display: block;
}
a.menucontact{
   width: 80px;
   height: 30px;
   background-image: url(images/menu_contact_automne.gif);
   display: block;
}
```

Une marge est fixée à gauche, de 20 pixels, pour mettre un peu d'espace avant chaque élément du menu. Cette marge est commandée par la propriété `margin: 0 0 0 20px` dans le sélecteur `li`.

Le menu est doublé par un autre menu vertical, qui explique un peu chaque rubrique. Le deuxième menu vertical est formé d'une colonne (une boîte) dans lequel s'échelonne un ensemble de boîtes `contenulignehome`. Les boîtes s'empilent les unes sur les autres et chacune contient elle-même deux boîtes. La première des boîtes, `contenubande`, a un fond d'une couleur légèrement plus claire, qui crée une bande verticale, tandis que la deuxième boîte, `contenuhome`, contient le commentaire sur la rubrique (voir Figure 5.11).

Cette disposition permet à chaque paire de boîtes d'être parfaitement alignée.



▲ Figure 5.11 : *Le menu vertical*

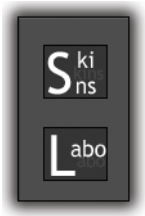
```
.contenuhome{
   margin-left: 95px;
   height: 100%;
   text-align: justify;
}
.contenulignehome{
   margin-bottom: 20px;
   width: 450px;
   float: left;
}
.contenubande{
   width: 85px;
   text-align: center;
   font-weight: bold;
   color: #DDD;
   float: left;
   height: 100%;
}
```

Le code XHTML se décline donc en une succession de boîtes qui se termine par le formulaire d'inscription à la lettre d'information.

```
<div class="conteneurdescriptionhome">
<div class="contenulignehome">
<div class="contenubande"><a href="skins.php">
<img src="images/home_skins_automne.gif" width="50px"
```

```
height="42px" alt="Choose your way !" title="Choose your
way !" border="0" /></a></div>
<div class="contenuhome">Parce que…</div>
         </div>
   <div class="contenulignehome">
   <div class="contenubande"><a href="labo.php">
<img src="images/home_labo_automne.gif" width="50px"
height="42px" alt="Labo" title="Labo" border="0" /></a></div>
   <div class="contenuhome">Dans cet espace…</div>
         </div>
   <div class="contenulignehome">
       <div class="contenubande">
<img src="images/home_portfolio_automne.gif" width="50px"
height="42px" alt="Portfolio" title="Portfolio" border="0" />
</div>
      <div class="contenuhome">Vous pourrez…</div>
         </div>
   <div class="contenulignehome">
      <div class="contenubande"><a href="webmaster.php">
<img src="images/home_webmaster_automne.gif" width="50px"
height="42px" alt="Webmaster" title="Webmaster" border="0" />
</a></div>
          <div class="contenuhome">Ici…</div>
         </div>
         <div class="contenulignehome">
          <div class="contenubande"><a href="humeurs.php">
<img src="images/home_humeurs_automne.gif" width="50px"
height="42px" alt="Humeurs" title="Humeurs" border="0" />
</a></div>
<div class="contenuhome">Cette rubrique… </div>
   </div>
   <div class="contenulignehome">
 <div class="contenubande"><a href="liens.php">
   <img src="images/home_liens_automne.gif"
        width="50px" height="42px"
        alt="Liens" title="Liens" border="0" /></a></div>
   <div class="contenuhome">Une petite liste…</div>
   </div>
<div class="contenulignehome">
<div class="contenubande"></div>
   <div class="contenuhome">
   <form method="post" action="php/newsletter/validemail.php">
 <label for="email">Newsletter :
<input type="text" name="email" size="20px" /></label>
<input type="submit" name="Submit" value="S'inscrire" />
       </form>
         </div>
         </div>
         </div>
```

Les images de liens sont ici au milieu de la boîte de colonne. Ici les images sont de vraies images et ne sont pas simplement des images de fond. Elles sont traitées par les navigateurs comme du texte et obéissent donc à la commande `text-align: center`. La position est flottante à gauche.



◄ Figure 5.12 : *Quelques boîtes*

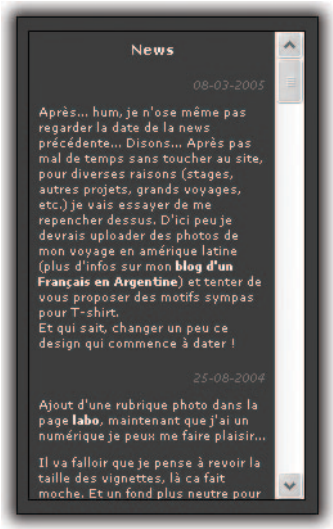
Le conteneur qui contient toute cette partie a comme propriété `height:` 100% qui lui permet de descendre jusqu'au bas de la page.

```
.conteneurdescriptionhome{
   float:left;
}
.contenuhome{
   margin-left: 95px;
   height: 100%;
   text-align: justify;
}
.contenulignehome{
   margin-bottom: 20px;
   width: 450px;
   float: left;
}
```

La boîte de news flotte, elle aussi, à gauche avec une marge de 40 pixels à gauche et de 20 pixels en bas. La hauteur est définie à 200 pixels et la propriété `overflow` arrêtée à `auto` permet à cette boîte d'afficher un ascenseur dès que le texte est trop grand (voir Figure 5.13).

```
.conteneurnews{
   width: 200px;
   height: 350px;
   overflow: auto;
   border: 1px solid #000000;
   float: left;
   margin: 0 0 20px 40px;
```

```
    padding: 7px;
}
```



◄ Figure 5.13 : *La boîte de nouvelles*

La boîte en bas, avec le style et la propriété `clear: both`, ajoute un espace après chaque news.

```
<div class="conteneurnews"><div class="titrenews">News</div>
      <div class="datenews">08-03-2005</div>
      <div class="textenews"><p>Après… </p></div>
      <br /></div>
      <div style="clear: both;"></div>
   </div>
```

Ce site est bien conçu. Le graphiste a fait preuve d'une bonne anticipation des besoins futurs du site et montre un réel souci de communiquer.

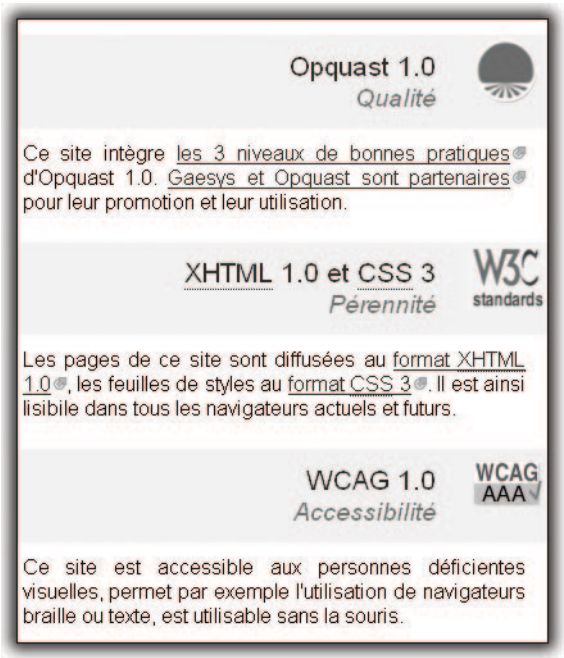
Clair et corporate

Le site gaesys.com est l'exemple type d'un site corporate (entreprise), avec son fond blanc, qui donne le meilleur contraste pour les différents textes ou liens qui parsèment la page d'accueil. Il est d'autant plus "corporate" qu'il est "B to B" (c'est-à-dire qu'il s'adresse à d'autres

entreprises). Le logo lui-même reflète parfaitement la philosophie de cette société, qui propose de la communication à d'autres sociétés et les outils web qui en sont les vecteurs.

▲ Figure 5.14 : *gaesys.com*

Les sites que gaesys.com propose de faire pour d'autres sociétés sont déjà aux normes Opquast 1.0, XHTML 1.0 et CSS 3, et WCAG 1.0. Gaesys a d'autant plus à cœur de s'y conformer pour son propre site, qui est une vitrine de son travail (voir Figure 5.15).



▲ Figure 5.15 : *Les normes*

La norme Opquast

La norme Opquast propose une liste des 153 bonnes pratiques (`www.opquast.com/bonnes-pratiques/liste/`), qui est un code de bonne conduite de la réalisation de sites web concernant aussi bien la communication, le code, les conventions, le serveur, etc.

La feuille de style commence par mettre les compteurs à 0, c'est-à-dire à remettre les marges internes et externes à 0. Ces marges sont créées par défaut sur certains navigateurs.

```
/* RAZ pref browser */
* {
   margin: 0;
   padding: 0;
}
```

À l'instar du site précédent, le code est bien structuré sur cette feuille de style, allant du général au particulier, agrémenté de commentaires et de claires séparations entre chaque partie.

Cette feuille s'occupe d'Internet Explorer, qui est le navigateur le plus répandu actuellement, mais qui obéit peu aux normes en vigueur. Le développeur, ici, tient compte des différentes aberrations de ce navigateur en insérant des codes que seul Internet Explorer sera à même de comprendre.

Pour en savoir plus sur les aberrations d'Internet Explorer, consultez le chapitre *La réalisation du site*.

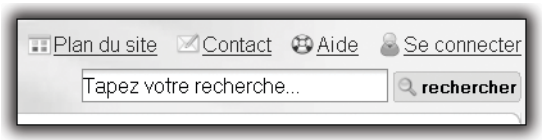
```
/* Base */
html {
   font-family: sans-serif;
   font-size: 80%;
}
body {
   padding-top: 0.6em;
   text-align: center;    /* Compat IE5 */
   background: #EBF4FD url("/images/main_background.jpg")
no-repeat fixed;
}
#structure { /* Compat IE5 */
   width: 90%;
   max-width: 960px;
   text-align: left;
   margin-left: auto;
   margin-right: auto;
}
```

La boîte de structure est à l'intérieur de `body` et en épouse exactement ses limites. Cette boîte contient les boîtes de classe `header` et `body`. Elle comble les failles d'Internet Explorer, comme le commentaire l'explique. Internet Explorer applique `text-align` aux boîtes et aux textes qui y sont contenus, tout en refusant `margin: auto`. Ici le développeur commence par positionner la boîte au centre pour Internet Explorer, dans le sélecteur `body` avec `text-align: center`, puis rectifie avec `margin: auto` et `text-align: left` dans le sélecteur `structure`, qui sert de double à la balise `body`. Notez que `max-width` ne s'appliquera pas à Internet Explorer.

La page est donc constituée d'une en-tête, d'un pied de page et d'un cartouche blanc délimitant le contenu.

L'en-tête regroupe un menu principal et un formulaire de recherche.

```
#gestion input[type="submit"], #gestion .submit {
   background: #d1e2f3 url("/images/icons/search.png")
no-repeat 2px center;
   padding-left: 18px;
}
```



▲ Figure 5.16 : *Petit menu*

L'image de fond est une loupe qui représente la fonction de recherche.

```
<div class="module" id="gestion">
<ul class="menu" id="menu1">
<li id="sitemap">
<a href="/sitemap" title="Accéder à la page Plan du site"
accesskey="3">Plan du site</a>
 </li>
 <li id="contact">
    <a href="/messages/" title="Accéder à la page Contact"
accesskey= "6">Contact</a>
</li>
<li id="help">
<a href="/help/" title="Accéder à la page Aide" accesskey="0">
Aide</a>
</li>
<li id="login">
<a href="/private/" title="Accéder à la page Se connecter"
accesskey="5">Se connecter</a>
</li>
</ul>
 <form action="/research" method="get">
<fieldset>
<label for="search">Rechercher sur ce site</label>
<input accesskey="4" type="text" name="searchstring"
value="Tapez votre recherche…" size="30" id="search" />
<input class="submit" type="submit" value="rechercher" />
</fieldset>
```

```
</form>
</div>
```

Le site propose aux internautes des touches d'accès pour accéder aux liens par un clavier. La propriété `accesskey` peut s'insérer dans les balises `a` ou `input` d'un formulaire. Le visiteur peut alors appuyer sur l'une des touches en question pour accéder à la page derrière le lien ou à la fonction représentée par le formulaire. Les différentes modalités sont expliquées dans l'aide et facilitent l'accès du site aux personnes qui naviguent sur Internet avec des navigateurs audio.

▲ Figure 5.17 : *Aide*

L'en-tête a un fond transparent et une image de fond en bas de sa boîte. Cette image de fond sert de bordure au cartouche situé dessous et lui fournit une bordure à angles ronds.

```
#header {
   background: transparent url("/images/toproundpng")
no-repeat right bottom;
}
#header .hr {
    padding-top: 1em !important;
   background: transparent url("/images/toproundpng")
no-repeat left bottom;
}
```

Les angles ronds sont obtenus par la balise `header` et par une classe `hr` à l'intérieur de cette balise. La propriété `padding-top` accorde un peu d'espace entre le formulaire dans l'en-tête et le cartouche de contenu.

▲ Figure 5.18 : *Et ronds et ronds…*

Le cartouche blanc à angles ronds est posé sur un fond bleu et blanc dont le haut figure des touches de clavier. Le motif d'arrière-plan descend lorsque le visiteur utilise l'ascenseur, grâce à la valeur `fixed` mise dans le sélecteur `body`. Les couleurs employées sont celles de Windows XP par défaut, ce qui est remarquable. C'est le même bleu que l'ascenseur, par défaut.

▲ Figure 5.19 : *L'image de fond*

On dirait que l'entreprise cherche à se fondre, à se faire discrète pour être d'autant mieux au service du client.

```
#body {
   background: white;
   border-left: 1px solid #b4c2cc;
   border-right: 1px solid #b4c2cc;
   clear: both;
}
#body hr {
   clear: both;
   border: 1px solid white;
   width: 95%;
   margin: 0px !important;
   margin-left: 1% !important;
}
#footer {
   padding-bottom: 10px;
   margin-top: -1px;
   font-size: 0.9em;
   color: #025;
   clear: both;
   background: transparent url("/images/bottomround.png")
no-repeat right top;
}
#footer .hr {
    height: 8px;
   background: transparent url("/images/bottomround.png")
no-repeat left top;
   margin-bottom: 1em;
}
.hr {
   width: 40%;
   clear: both;
}
```

L'image des coins ronds est posée par paires. Le cartouche est conçu pour faire au maximum 960 pixels. Par conséquent, les images de bordure supérieure et de bordure inférieure font au maximum 960 pixels. Elles sont formées de la ligne de bordure et des deux angles ronds de chaque côté, pour rejoindre les bordures droite et gauche qui, elles, sont des créations CSS. Pourquoi insérer deux images à chaque bordure horizontale ? Tout simplement pour prévoir le cas où le cartouche, qui fait 90 % du tout, serait plus petit et où l'angle de l'extrémité serait caché. Les deux images forment une sorte d'image télescopique. Comme on peut le constater, dans le cartouche `header`, l'image est placée en bas à droite,

puis dans la classe `hr` de `header`, l'image est placée en bas à gauche. Il en est de même pour le cartouche `footer`, bâti exactement de la même façon, excepté que les images sont en haut de la boîte.

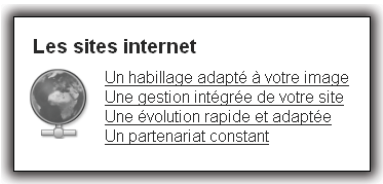
Le cartouche est formé de plusieurs parties : en haut à gauche, l'arborescence de la page courante, et un formulaire à droite, qui permet d'imprimer la page ou d'en modifier les polices. Le corps de la page lui-même est formé d'une boîte qui couvre toute la largeur du cartouche. À la fin de cette boîte apparaît la classe `hr`, qui permet de repousser les boîtes à droite et à gauche avec la propriété `clear`.

◄ Figure 5.20 :
Petit menu à droite

En dessous, une boîte est affichée, contenant deux boîtes côte à côte, puis deux boîtes flottantes (l'une à gauche, l'autre à droite). Enfin, une boîte couvre la dernière partie sur toute la largeur. La boîte de classe `hr` est savamment distribuée sur la page pour que tout s'ordonne. Quelques parties sont soulignées d'une couleur claire qui accroche le regard au premier examen de la page. Ce sont les boîtes contenant l'éditorial et celle qui mène vers les réalisations de sites web déjà abouties.

```
<div class="resumes">
<div>
<h2>Les sites internet</h2>
<a href="/prestations/internet" title="Les sites internet">
<img src="/images/icons/internet.png" alt="Les sites internet"
height="70" width="70" /></a>
 <ul>
    <li><a href="/prestations/internet#habi"
title="Un habillage adapté à votre image">Un habillage
adapté à votre image</a></li>
 <li><a href="/prestations/internet#gest" title="Une gestion
intégrée de votre site">
Une gestion intégrée de votre site</a></li>
<li><a href="/prestations/internet#evol"
title="Une évolution rapide et adaptée">
Une évolution rapide et adaptée</a></li>
<li><a href="/prestations/internet#part"
title="Un partenariat constant">
Un partenariat constant</a></li>
```

```
 </ul></div>
<div>
<h2>Les intranets de Gestion</h2>
<a href="/prestations/intranet"
title="Les intranets de Gestion">
<img src="/images/icons/intranet.png"
alt="Les intranets de Gestion" height="70"
width="70" /></a>
 <ul>
<li><a href="/prestations/intranet#orga"
title="L'organisation de vos données">
L'organisation de vos
 données</a></li>
<li><a href="/prestations/intranet#solu"
title="Des solutions a vos besoins métiers">
Des solutions a vos besoins métiers</a></li>
<li><a href="/prestations/intranet#inte"
title="L'intégration de vos processus">L'intégration
de vos processus</a></li>
<li><a href="/prestations/intranet#rela"
title="Vos relations facilitées.">Vos relations facilitées</a>
</li></ul>
</div>
</div>
```



◄ Figure 5.21 : *La boîte de gauche*

Ici les deux boîtes sont simplement posées côte à côte sans aucune forme, ni intervention de style quelconque. Ensuite intervient la boîte nommée `hr`, qui fait le vide de chaque côté et sépare les deux boîtes, flottant à gauche et à droite, qui suivent.

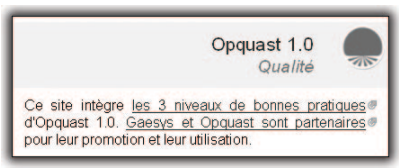
◄ Figure 5.22 : *La boîte de droite*

```
<div class="hr"></div>
```

La boîte gauche contient elle-même plusieurs boîtes. Le conteneur `hr` évite que des boîtes ne viennent se glisser sur le côté et ajoute aussi un espace de séparation.

```
<div class="left">
<div class="hsection" id="edito">
<h2>L'investissement durable</h2>
 <p>Nous avons choisi… </p>
<a class="more" href="durable/"
title="L'investissement durable">
En savoir plus sur l'investissement durable</a>
 </div>
 <div class="hsection" id="introenga">
 <h2>Notre engagement</h2>
 <p>Lorsque nous…</p>
<a class="more" href="engagement/" title="Notre engagement">
En savoir plus sur notre engagement</a>
<div class="hr"></div>
</div>
<div class="real">
<p><a href="/realisations/"
title="Nos réalisations, problématiques et solutions">
Nos derniers projets réalisés</a>,
sites internet, boutiques, et intranet. </p>
 </div>
 </div>
```

La boîte droite affiche les normes. Elles sont affichées à cet endroit car elles représentent une information secondaire par rapport à l'information plus importante sur la partie gauche. Les images affichées sont des images de fond dans des boîtes qui ont une classe et une identification. La classe permet de caractériser toutes les boîtes apparentées tandis que l'identification permet de les individualiser. Le principe ici est simple : l'image de fond est placée en haut à droite et les titres ont un `padding-right` un peu plus grand que la largeur de l'image, c'est-à-dire que le titre va se placer à gauche de l'image, à 90 pixels du bord.



◄ Figure 5.23 : *La boîte Opquast*

```
<div id="engagement">
 <div class="bzone"></div>
 <div class="zone" id="opquast">
<div class="title">
Opquast 1.0
 <div>Qualité</div>
</div>
 <p>Ce site intègre
<a href="http://opquast.com/declarations/?site=705" title="…">
les 3 niveaux de bonnes pratiques</a> d'Opquast 1.0.
<a href="http://www.opquast.com/partenaires/"
title="les partenaires du projet Opquast">Gaesys et Opquast
sont partenaires</a> pour leur promotion et leur utilisation.
</p> </div>
 <div class="zone" id="w3c">
 <div class="title">
 <acronym title="eXtanced Hypertext Markup Language">
XHTML</acronym>
 1.0 et <acronym title="Cascading Style Sheet">CSS</acronym> 3
 <div>
 Pérennité
 </div>
 </div>
 <p>
 Les pages de ce site sont diffusées au <a href=
 "http://validator.w3.org/check?uri=http://www.gaesys.com/"
title="Site valide XHTML 1.0">format <acronym title=
 "eXtensed Hypertext Markup Language">XHTML</acronym> 1.0</a>,
les feuilles de styles au <a href=
 http://… title="Site valide CSS3">format
<acronym title="Cascading Style Sheet">CSS</acronym> 3</a>.
Il est ainsi lisible
 dans tous les navigateurs actuels et futurs.
 </p>
</div>
 <div class="zone" id="wcag">
 <div class="title">
 WCAG 1.0
 <div>
 Accessibilité
 </div>
 </div>
 <p>
 Ce site est accessible…
 </p>
</div>
 <div class="hr"></div>
 </div>
 <div class="isection" id="avis">
```

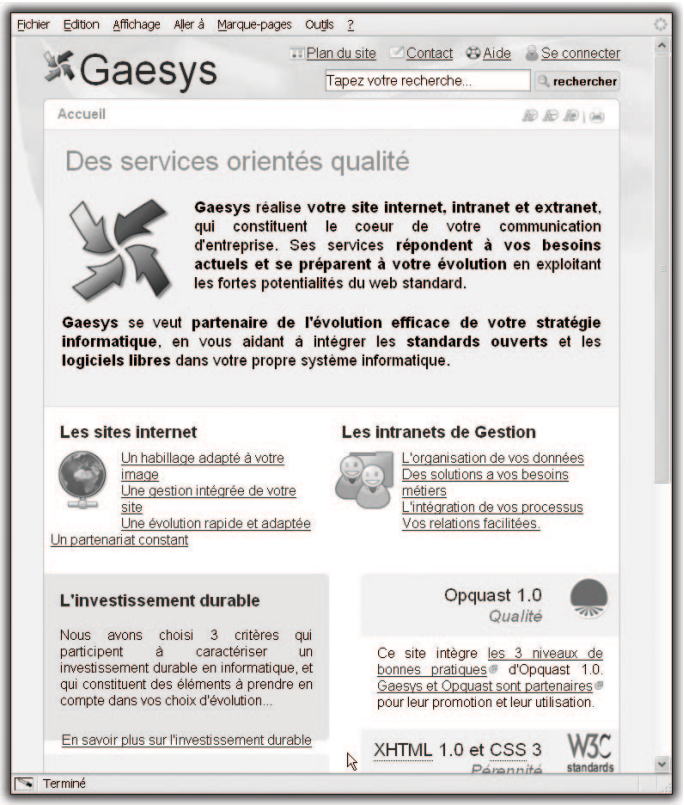
```
 <p>
 <strong>Votre avis nous intéresse !</strong>…
 </p>
</div>
 </div>
 <hr />
</div>
```

La balise `acronym` permet au navigateur d'afficher un drapeau qui déploie le libellé complet de l'acronyme au passage de la souris. Encore une fois, la balise nommée `hr` arrive à point pour fermer ces boîtes et ouvrir la voie à la dernière boîte du cartouche, nommée `avis`.

```
#opquast {
background: #ebf4fd url("/images/opquast.png")
no-repeat right top;
}
#w3c {
background: #ebf4fd url("/images/w3c.png")
no-repeat top right;
}
#wcag {
    background: #ebf4fd url("/images/wcag.png")
no-repeat top right;
}
#engagement .title div {
   float: none;
   font-size: 0.9em;
   font-style: italic;
   color: #4477AA;
}
#engagement {
   float: right;
   width: 45%;
   background-image: none;
}
#engagement2 .title div {
   font-size: 0.9em;
   font-style: italic;
   color: #4477AA;
}
.left2 {
   width: 54%;
   float: left;
}
#engagement2 {
   width: 45%;
   float: right;
   background: transparent url("/images/charte.png")
```

```
no-repeat top left;
}
engagement .zone {
   background-color: #ebf4fd;
}
#engagement2 .zone {
   margin: 0 !important;
   padding: 0 !important;
}
#engagement2 .zone .title {
   font-family: Tahoma;
   font-size: 18pt;
   text-align: right;
   color: #112233;
   height: 60px;
   margin: 0;
   padding: 0;
   padding-right: 90px;
   padding-top: 10px;
}
#engagement2 .zone .title div {
   padding-left: 1em;
   font-size: 12pt;
}
#engagement2 .zone p {
   margin: 0 !important;
   padding: 0;
   padding-top: 5px;
   height: 65px;
}
#engagement .zone .title {
   background: transparent url("/images/topleftblue.png")
no-repeat top left;
   font-size: 1.3em;
   padding-right: 90px;
   text-align: right;
   color: #003366;
   padding-top: 0.6em;
}
#engagement .zone .title div {
   padding-left: 1em;
   padding-bottom: 0.6em;
   background: transparent url("/images/leftblue.png")
no-repeat bottom left;
}
#engagement .zone p {
   background: #FFFFFF;
   padding-top: 0.6em !important;
}
```

Le point remarquable dans la construction des styles de la page est que la distribution est cohérente entre les dimensions relatives aux images (et à tout ce qui s'y rattache), qui sont formulées en pixels, et celles des autres boîtes, qui sont en em, c'est-à-dire en une dimension relative.



▲ Figure 5.24 : *Réduction de taille 1*

Ainsi, la réduction de la résolution de l'écran ou de la taille de la fenêtre du navigateur ne défait pas cette savante architecture, qui est remarquablement élastique. L'agrandissement des caractères dans un navigateur ne nuit pas, non plus, à l'harmonie ou à la lisibilité de l'ensemble.

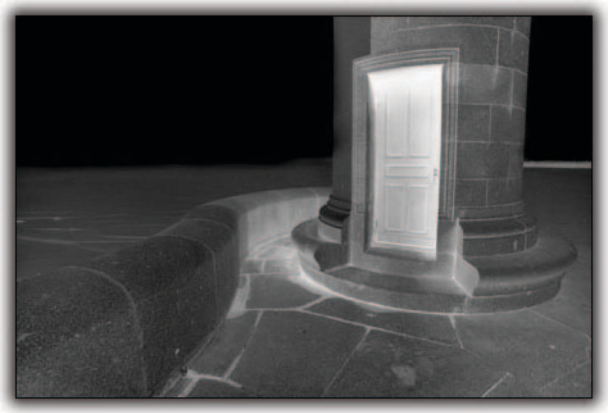
◄ Figure 5.25 : *Réduction de taille 2*

!important

La valeur `!important` permet au développeur de rendre une propriété prioritaire sur la feuille de style par défaut du navigateur et sur celle de l'utilisateur. Cette propriété a priorité sur tout, excepté sur une propriété qui serait étiquetée `!important` dans une feuille de style de l'utilisateur.

5.2 Atelier d'artiste

Mikaël Helleux est photographe. Son site `http://etc.photo.free.fr/` lui permet d'exposer ses œuvres. Ses pages sont suffisamment sobres pour mettre en valeur ses photos, qu'elles soient en couleur ou en noir et blanc.



▲ Figure 5.26 : *Arrivez*

La première page du site commence par une astuce qui permet de transformer la photo du positif au négatif lorsque la souris passe sur le lien en légende. La photo représente la porte d'un phare.

```
#index_photo {
   display:block;
   width:448px;
   height:297px;
   border:1px solid #808080;
   background:url(images/bazar/phare_nega.jpg)
no-repeat 50% 0%;
}
#index_photo img {
   filter:alpha(opacity=1);
   -moz-opacity:0.1;
}
```

L'auteur a utilisé une boîte avec une image de fond dans laquelle il a inséré une photo. L'image de fond est en négatif et l'image de "surface" est en positif, toutes les deux en noir et blanc. Pour que l'image de fond apparaisse, la photo de surface est affectée d'une transparence (voir Figure 5.27).

Il s'agit d'une propriété CSS3, ce qui oblige l'auteur à adjoindre du code JavaScript. Le JavaScript permet, au passage de la souris sur le lien, de

modifier l'opacité de la photo de surface, qui devient transparente, et de révéler ainsi la photo de fond.

▲ Figure 5.27 : *Entrez*

La propriété `filter` appelle des effets. Paradoxalement, ils sont acceptés par Internet Explorer 6, mais pas par Firefox 1.5.

Le filtre `alpha` utilisé ici a une transparence maximale, attribuée avec `opacity=1`.

5.3 Un site fédérateur

Le site `www.fleurs-livraison.info` propose au visiteur des offres de plusieurs sites de fleurs sur Internet et de prestataires de livraison, avec un large éventail de bouquets ou de budgets. Il fait office d'intermédiaire entre les sites de livraison de fleurs et l'acheteur (voir Figure 5.28).

La ligne graphique est claire et la feuille de style bien ordonnée et intelligente.

▲ Figure 5.28 : *La page d'accueil*

L'auteur fait d'abord table rase des réglages des navigateurs par défaut.

```
body {
margin:0;
padding:0;
background:#fff;
font:1em Arial, Helvetica, sans-serif
}
body {
text-align:center
}
/** Centrage de la page dans ie **/
```

Les marges sont remises à 0 et le centrage des blocs qui, sur Internet Explorer, se fait avec `text-align`, est assuré. Ce centrage signifie que, un peu plus loin, il faudra rectifier et ajouter `text-align: left`, une fois que le centrage des blocs pour les navigateurs conformes aura été indiqué.

```
#master {
margin:auto;
text-align:left;
width:800px;
}
```

Le conteneur `master` épouse le contenu de la balise `corps` et permet de rectifier le bidouillage (vu plus haut à propos d'Internet Explorer). `text-align: center` a permis de centrer le conteneur pour Internet Explorer, `margin: auto` le fait pour les autres navigateurs, puis `text-align: left` remet le texte à gauche pour Internet Explorer et les autres navigateurs. Cette manière de faire complique le code et n'est pas conforme. Mais les développeurs doivent tenir compte de la non-conformité de certains navigateurs.

```
<body>
<div id="master">
```

La stratégie architecturale de la page est assez simple :

- un conteneur pour l'en-tête avec la bannière comme image de fond ;
- un conteneur pour la partie centrale ;
  - un conteneur pour la navigation ;
  - un conteneur pour le contenu propre à la page ;
- un conteneur pour le pied de page.

Pour que les deux colonnes du contenu central, prises en sandwich entre l'en-tête et le pied, puissent se terminer proprement afin de faire place nette et de laisser de l'espace au pied de page, une balise `br` avec le nom `clear` comme classe est incluse à la fin de la deuxième colonne.

```
/** Utiles **/
.clear {clear:both;}
```

Tous les conteneurs de gauche sont tracés au cordeau car le motif de fond est un rectangle répété de haut en bas (`repeat-y`). Cette ligne a une

bordure latérale qui lui donne son apparence de relief. Le conteneur est coiffé d'une balise h2 avec, comme image de fond, le haut arrondi du cartouche, tandis que le bas est rempli d'une classe avec l'image de fond du bas du cartouche.

◄ Figure 5.29 : *Un conteneur de gauche*

```
<!-- Menu Les boutiques -->
<div id="menu2">
<h2>Les fleuristes</h2>
<ul>
<li><a href="livraison/fleurs_aquarelle.asp"
title="Livraison fleurs - Aquarelle">Aquarelle</a></li>
   <li><a href="livraison/fleurs_au_nom_de_la_rose.asp"
title="Livraison fleurs - Au nom de la rose">Au nom de la rose
</a></li>
<li><a href="livraison/fleurs_bebloom.asp"
title="Livraison fleurs - Bebloom">Bebloom</a></li>
<li><a href="livraison/fleurs_florajet.asp"
title="Livraison fleurs - Florajet">Florajet</a></li>
<li><a href="livraison/fleurs_interflora.asp"
title="Livraison fleurs - Interflora">Interflora</a></li>
<li><a href="livraison/fleurs_bouquet_nantais.asp"
title="Livraison fleurs - Le bouquet Nantais">
Le bouquet Nantais</a></li>
<li><a href="livraison/pres_de_chez_vous.asp"
title="Livraison fleurs - Près de chez vous">
Près de chez vous</a></li>
</ul>
<span class="menu2bottom"></span>
</div>
<!-- Fin colonne de gauche -->
```

La feuille de style, elle, ajoute les images.

◄ Figure 5.30 :
La coiffe du cartouche "Les bouquets"

◄ Figure 5.31 :
Le corps du cartouche "Les bouquets"

◄ Figure 5.32 :
Le pied du cartouche "Les bouquets"

```
/** Menus **/
#menu1 {width: 190px; background: url(images/bg_menu1.gif)
repeat-y; margin-bottom:0px}
#menu2 {width: 190px; background: url(images/bg_menu2.gif)
repeat-y;}

#zonemenu h2 {padding:11px 10px 12px 10px;}
#menu1 h2 {background: url(images/bg_menu1-top.gif)
no-repeat;}
#menu2 h2 {background: url(images/bg_menu2-top.gif)
no-repeat;}

#zonemenu ul, #zonemenu li {margin:0; padding:0;
list-style: none;}
#menu1 ul {width:100%; background: #0099CC
url(images/bg_menu1.gif)
repeat-y;}
#menu2 ul {width:100%; background: #0099CC
url(images/bg_menu2.gif)
repeat-y;}

#zonemenu li {margin:0 4px 4px 4px;}
#menu1 li {background:#65BBD2;}
#menu2 li {background:#B6B6B6;}

#zonemenu a {font-size:0.8em; color:#fff; display:block;
padding:2px 4px 4px 34px; margin-top:-2px;}
#menu1 a {background: url(images/menu1-puce.gif) no-repeat;
font-weight: bold; color:#306376;}
#menu2 a {background: url(images/menu2-puce.gif) no-repeat;
color: #666666; font-weight: bold;}
#menu1 a:hover {background: url(images/menu1-puce.gif)
no-repeat; font-weight: bold; color:#ffffff;}
#menu2 a:hover {background: url(images/menu2-puce.gif)
no-repeat; color: #ffffff; font-weight: bold;}
```

```
.menu1bottom, .menu2bottom {height:15px; width:190px;
margin:0; display:block}
.menu1bottom {background: url(images/bg_menu1-bas.gif)
no-repeat bottom;}
.menu2bottom {background: url(images/bg_menu2-bas.gif)
no-repeat bottom;}
```

Les menus déclinent comme puce un carré encadrant une flèche, qui est une image de fond, et comme couleur de fond la même que celle du cartouche.

◄ Figure 5.33 : *La flèche*

La balise `li` a comme propriété `display: block`. Ainsi, tout "survol" du conteneur `li` provoque le changement de couleur du lien. Le `padding` agit sur le retrait de la puce.

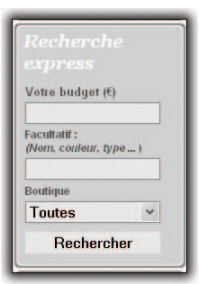
◄ Figure 5.34 : *Sur un "survol"…*

Le formulaire de recherche est remarquable par son code XHTML et sa feuille de style.

```
/** Formulaire de recherche **/
#frm-recherche {
width: 190px;
background: url(images/bg_recherche.gif) repeat-y;
margin-bottom:0px;
}
#frm-recherche h2 {
background: url(images/bg_recherche-top.gif) no-repeat;
font-size:1em;
font-style:italic;
padding:11px 10px 12px 10px;
}
#frm-recherche form {
margin:0;
```

```
padding:0;
}
#frm-recherche p {
margin: 0 12px
}
#frm-recherche label {
font-size:0.9em;
}
#frm-recherche input, #frm-recherche select {
margin:4px 0;
width:150px;
background-color:#EEEDC4;
border:0;
border:1px solid #948E1C;
}
#frm-recherche .button {
border:1px solid #eaeaea;
margin-top:6px;
}
.frmrecherche-bottom {
background: url(images/bg_recherche-bas.gif) no-repeat bottom;
height:15px;
width:190px;
margin:0;
display:block
}
```

Le cartouche du formulaire de recherche est réalisé de la même façon que les deux autres, avec le haut et le bas qui terminent la bordure avec des angles ronds.



◄ Figure 5.35 :
Le cartouche de recherche

À l'intérieur du formulaire, les champs de texte ou du menu sont d'une couleur légèrement plus claire que le fond `background-color:#EEEDC4` et que le bouton de soumission du formulaire, qui contient la propriété `input`.

```
<!-- Formulaire de recherche -->
<div id="frm-recherche">
<h2>Recherche express</h2>
<form action=" fleurs_recherche_express.asp" method="get">
<p><label for="budget">
<span class="Style10">Votre budget (&euro;)</span>
</label></p>
<p class="Style10">
   <input name="budget" id="budget" type="text" tabindex="1"
maxlength="5"></p>
<p class="Style10">
   <label for="Facultatif"> Facultatif :<br>
   <em>(Nom, couleur, type…</em> ) </label>
</p>
<p class="Style10">
<input autocomplete="off" name="facultatif" id="facultatif"
type="text" onkeyup="declenche(this);" maxlength="50">
</p>
<p class="Style10">
<div align="left" id="resultat" name="resultat" class="ajax">
</div></p>
<p class="Style9">
<span class="Style10">
<label for="boutique">Boutique</label>
</span>
<label for="boutique"></label>
</p>
<p> <span class="Style9">
<select name="boutique" class="select" id="boutique">
   <option value="0" selected> Toutes</option>
<option value="1">Aquarelle</option>
<select>
<p>
<input class="button" type="submit" name="Submit"
value="Rechercher" id="Submit">
</p>
</form>
<span class="frmrecherche-bottom"></span>
</div>
```

La colonne de droite affiche le contenu de la page. Elle est réalisée de la même manière que la colonne précédente avec trois conteneurs. Le troisième conteneur intègre une image à gauche, avec le texte aligné à droite et un `padding` de 13 pixels ajouté à la largeur de l'image.

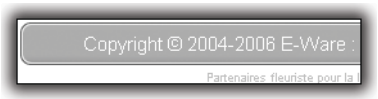
▲ Figure 5.36 : *La colonne "de droite"*

```
.annoncespl p{padding-left:146px}
```

Le XHTML s'égrène selon le même modèle que précédemment.

```
<div class="annoncespl">
<span class="annoncespltop"></span>
<img src="photo-accueil.jpg" alt="Femme avec roses rouge">
   <p>Offrir des…<br><br>En plus de…<br><br>
Vous pouvez…<br><br>Nous vous souhaitons d'avance…<br>
   </p><span class="annoncesplbas"></span>
</div>
```

Le pied de page se divise en deux parties horizontales. La première est dans un cartouche beige et la deuxième est un texte en petits caractères sur le fond blanc.



◄ Figure 5.37 : *le pied*

```
/** Pied **/
.piedtop {
width:800px;
height:8px;
display:block;
background: url(images/bg_pied-top.gif) no-repeat bottom;
margin-top:6px
}
#pied p.infos {
font-size:0.75em;
color:#fff;
text-align:center;
background: #00CC66 url(images/bg_pied.gif) repeat-y;
 }
#pied .infos a {color:#fff;}
#pied p.liensutiles {
color:#c0c0c0;
width:800px;
text-align:center;
margin:0;
padding-top:12px;
font-size:0.75em;
background: url(images/bg_pied-bas.gif) no-repeat left top;
font-size: 9px;
}
#pied .liensutiles a {
color:#c0c0c0;
text-decoration:none;
font-size: 9px;
}
```

L'image de fond est divisée en trois et distribuée dans les trois conteneurs du pied de page :

- le haut du cartouche ;
- le corps du cartouche ;
- le bas du cartouche placé en haut du dernier conteneur.

Fleurs-livraison est un joli site au service d'une bonne idée, à savoir regrouper toutes les offres de fleurs sur Internet, livraison comprise.

5.4 Construction du site

Auparavant, la seule manière de construire une page était de créer un tableau invisible avec des cellules dans lesquelles se positionnaient des images, des titres, des textes, des liens de navigation, le pied de page, le logo, la bannière de publicité. Le tableau est, en HTML, la seule balise permettant d'ordonner et de placer les éléments. Le HTML est conçu, au départ, pour afficher des textes avec des images qui sont prises comme des éléments de texte et qui s'insèrent là où le code est placé. La seule possibilité est d'aligner le texte sur l'image avec la propriété `align`. Celle-ci place le texte au bas, au milieu ou en haut de l'image, en prenant respectivement les valeurs `bottom`, `middle` et `top`.

La conception des calques (*layers*) révolutionne tout cela, avec Netscape le navigateur le plus utilisé à l'époque. Elle n'a jamais été validée par le consortium W3C. Viennent ensuite les balises `DIV` promues par Microsoft sur son navigateur Internet Explorer. Ces containeurs ont été validés. Sans CSS, ces boîtes sont disposées verticalement, les unes sur les autres, dans ce qu'on appelle le flux.

Le flux est la lecture ligne à ligne opérée par un navigateur quand il décode du HTML. Les premières données affichées sont écrites au début du fichier source. Les boîtes ont ce privilège de suivre le flux courant ou d'en sortir (voir Figure 5.38).

Le site classique est composé de trois colonnes, d'un en-tête et d'un pied, c'est-à-dire de cinq boîtes. Il s'agit ici de disposer ces boîtes dans un ensemble harmonieux et régulier. Ces cinq boîtes sont matérialisées par les conteneurs `DIV`.

Fichier Edition Affichage Aller à Marque-pages Outils ?
LE TITRE
LIENS GAUCHES LIENS GAUCHES LIENS GAUCHES LIENS GAUCHES LIENS GAUCHES LIENS GAUCHES LIENS GAUCHES LIENS GAUCHES LIENS GAUCHES
Le contenu Le contenu Le contenu

service de la conception de la page. Disposer d'un bon outil, c'est bien… encore faut-il savoir ce que l'on peut en faire.

▲ Figure 5.39 : *Un site avec découpe*

Une fois le dessin réalisé, le webmaster le découpe en code et en images. C'est là qu'il faut choisir le positionnement des colonnes et les imbrications de conteneurs.

Types de boîtes

Les CSS proposent deux types de boîtes :

- les boîtes en ligne (inline) ;
- les boîtes bloc.

Tab. 5.1 : Différences entes les boîtes bloc et les boîtes en ligne

Caractéristiques des boîtes bloc	Caractéristiques des boîtes en ligne
Elles sont disposées naturellement les unes au-dessus des autres.	Elles sont disposées naturellement les unes après les autres.

Tab. 5.1 : Différences entes les boîtes bloc et les boîtes en ligne

Caractéristiques des boîtes bloc	Caractéristiques des boîtes en ligne
Elles occupent toute la largeur du conteneur qui les contient.	Elles entourent horizontalement leur contenu.
Elles possèdent quatre marges externes (haut, bas, gauche, droite).	Elles possèdent deux marges externes (gauche, droite).
Elles sont insécables.	Elles sont sécables (pour aller à la ligne).
Elles peuvent seulement contenir d'autres boîtes en ligne.	Elles peuvent contenir tout type de boîte.
Il faut préciser la hauteur pour mettre de l'air autour du contenu.	Elles permettent de définir la taille des marges intérieures (`padding`).
Elles prennent en charge `span`, `em`, `a`, `img`, `br`, `i`, `strong`.	Elles prennent en charge `div`, `body`, `p`, `table`, `span`, `a`, `blockquote`, `h1–h6`, `ol`, `ul`, `li`, `hr`, `img`, `form`.

La frontière entre "en ligne" et "bloc" est poreuse

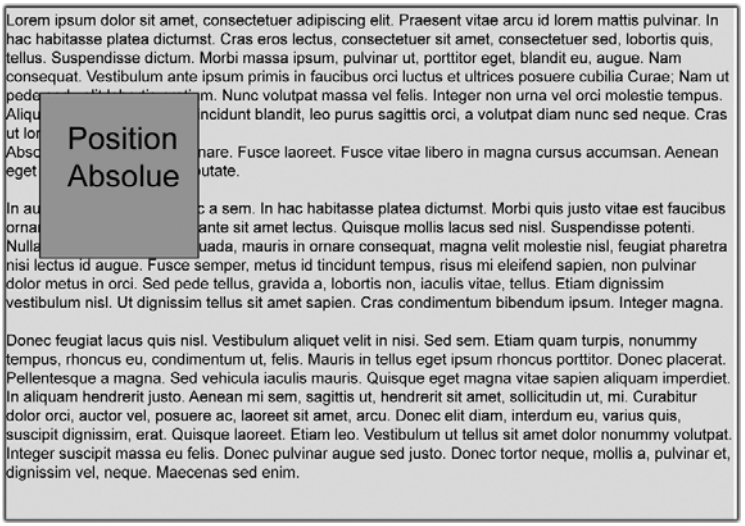
Avec la propriété `display: block` ou `display: inline`, il est possible de forcer le type de la boîte ou de l'affirmer pour certains navigateurs peu respectueux des normes W3C.

Position des boîtes

La position par défaut, `position: static;`, est dans le flux normal. Positionner une boîte signifie la sortir du flux naturel du HTML et la placer arbitrairement dans la page. La position peut être calculée par rapport à la page web (`absolute`), par rapport aux autres boîtes (`relative`) ou par rapport à l'écran (`fixed`).

absolute

Quand une boîte a une position absolue, sa place est définie par les propriétés `top`, `left`, `right`, `bottom`. Elles indiquent l'écart entre la boîte et le bord de la page web ou de l'élément qui la contient.



▲ Figure 5.40 : *Position absolue*

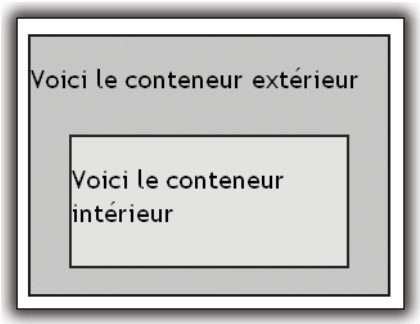
```
div#navigauche {
position: absolute;
left: 10px;
top: 120px;
}
```

Cette boîte sera positionnée à 120 pixels du bord supérieur du site et à 10 pixels du bord gauche, quelle que soit sa place dans le code HTML et quelle que soit la place des boîtes qui la contiennent.

```
<div id="fruit">
<p>Voici le conteneur extérieur</p>
<div id="noyau">
<p>Voici le conteneur intérieur</p>
</div>
</div>
```

Le code précédent affiche une boîte contenue elle-même dans une autre boîte. Le code CSS va bousculer ce joli ordonnancement en sortant la boîte `noyau` de sa place naturelle dans la boîte `fruit`.

```
#fruit {
background: #ACF;
border: 2px solid #00F;
height: 200px;
margin: 50px;
}
#noyau {
background: #FF0;
border: 2px solid #800;
height: 100px;
margin: 30px;
}
```

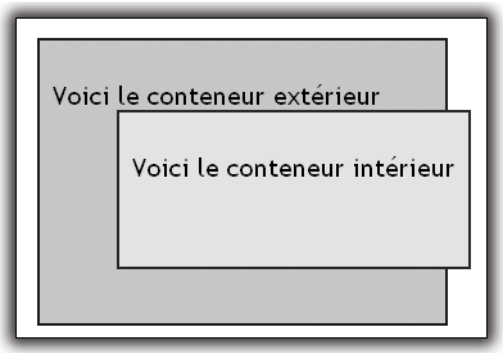


◄ Figure 5.41 : *Fruit et noyau 1*

Maintenant l'exercice va consister à extraire le noyau du fruit, c'est-à-dire à sortir le conteneur intérieur du conteneur extérieur avec la position `absolute`. La propriété `padding` évite que le texte soit collé au bord interne du conteneur. Le conteneur `noyau` est à 10 pixels du bord inférieur et à 10 pixels du bord droit.

```
#fruit {
background: #ACF;
border: 2px solid #00F;
height: 200px;
margin: 50px;
padding: 10px;
}
#noyau {
position: absolute;
bottom: 10px;
right:10px;
background: #FF0;
border: 2px solid #800;
height: 100px;
```

```
margin: 30px;
padding: 10px;
}
```



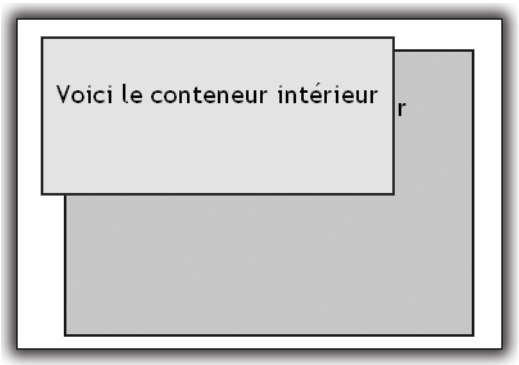
◄ Figure 5.42 : *Fruit et noyau 2*

Maintenant, le conteneur `noyau` mord sur le conteneur `fruit` avec une position absolue à 10 pixels du bord supérieur et du bord gauche.

(voir Figure 5.43)

```
#fruit {
background: #ACF;
border: 2px solid #00F;
height: 200px;
margin: 50px;
padding: 10px;
}
#noyau {
position: absolute;
top: 10px;
left:10px;
background: #FF0;
border: 2px solid #800;
height: 100px;
margin: 30px;
padding: 10px;
}
```

La place du conteneur `noyau` n'est pas réservée et donc le conteneur `fruit` n'est pas écarté et les deux conteneurs se retrouvent sur le même emplacement. Le deuxième conteneur, `noyau`, se retrouve au-dessus du conteneur `fruit` sans que le texte soit repoussé dans la partie visible.



◄ Figure 5.43 : *Fruit et noyau 3*

Si le texte s'allonge et si la boîte est dotée d'une hauteur précise, le texte débordera de la boîte. Si la hauteur n'est pas précisée, à travers la propriété `height`, la boîte va s'adapter à la longueur du texte.

overflow

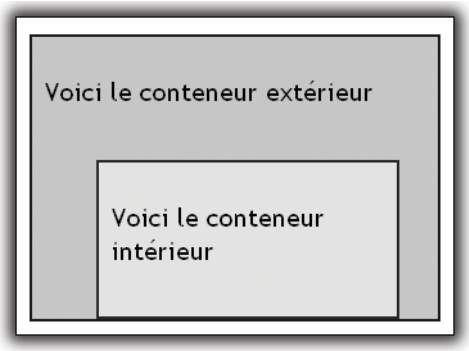
La propriété `overflow` gère le comportement du texte par rapport à la boîte qui le contient. Le comportement par défaut est `visible` : le texte déborde de la boîte si celle-ci est trop petite. Avec la valeur `hidden`, le texte reste contenu dans les limites de la boîte, mais aucun ascenseur ne permet de voir ce qui est caché. Avec `scroll`, le texte est contenu dans les limites de la boîte et est associé à une glissière quelle que soit la taille du texte contenu. `auto` provoque l'apparition de la glissière seulement si la taille du texte le demande.

relative

La position relative ne se calcule plus par rapport à la page du site mais en fonction de la place que l'élément devrait occuper en étant dans le flux normal. Le flux normal dans l'exemple précédent place le noyau dans le fruit. Reprenons l'exemple précédent :

```
#fruit {
background: #ACF;
border: 2px solid #00F;
```

```
height: 200px;
margin: 50px;
padding: 10px;
}
#noyau {
position: relative;
top: 10px;
left:10px;
background: #FF0;
border: 2px solid #800;
height: 100px;
margin: 30px;
padding: 10px;
}
```

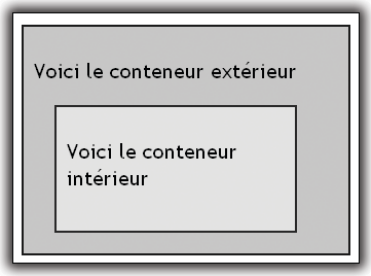


◄ Figure 5.44 : *Fruit et noyau 4*

Le conteneur `noyau` est en position relative par rapport au conteneur `fruit`, et non plus par rapport au bord de la page web. Il se trouve donc à 10 pixels du bord supérieur et à 10 pixels du bord gauche de l'élément qui le contient.

```
#fruit {
background: #ACF;
border: 2px solid #00F;
height: 200px;
margin: 50px;
padding: 10px;
}
#noyau {
position: relative;
bottom: 10px;
right: 10px;
background: #FF0;
```

```
border: 2px solid #800;
height: 100px;
margin: 30px;
padding: 10px;
}
```



◄ Figure 5.45 : *Fruit et noyau 5*

Comme précédemment, le texte pourra déborder si la hauteur spécifiée est trop faible, ou le noyau s'agrandir au détriment du fruit, si le texte est long. Dans tous les cas, la marge extérieure (`margin`) et la marge intérieure (`padding`) sont prises en compte.

fixed

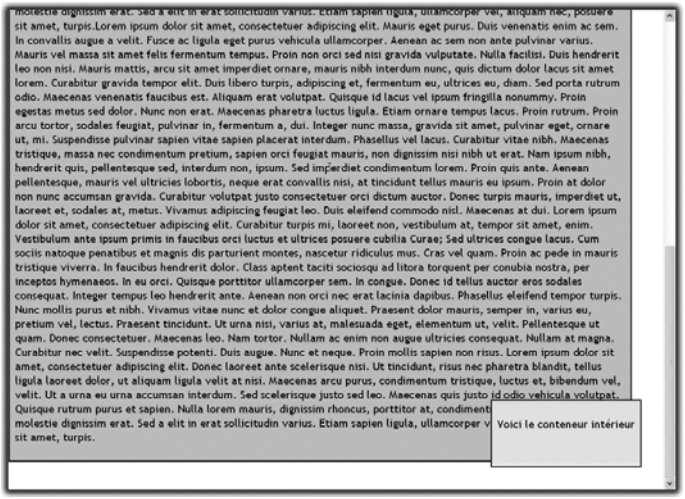
Une boîte de navigation transportant des liens de navigation peut avoir besoin, comme une palette flottante dans de nombreux logiciels graphiques, d'être à portée de clic de souris.

▲ Figure 5.46 : *Boîte en position fixée*

Pour peu que la page soit très haute, trois possibilités se présentent pour y avoir accès à partir du bas de la page :

- utiliser l'ascenseur ;
- disposer d'une ancre qui renvoie en haut de la page ;
- disposer d'une boîte, qui, pour n'être pas flottante, n'en est pas moins toujours rangée à la même place sur l'écran.

Le résultat n'est pas performant graphiquement, mais il est pratique pour l'utilisateur, à condition que cette boîte reste de dimension modeste.



▲ Figure 5.47 : *Même boîte, même position dans le navigateur*

Une page à trois colonnes avec la position relative

Pour faire une page en trois colonnes, le développeur pose des boîtes sur la page. En fait, elle contient classiquement cinq boîtes :

- la boîte d'en-tête qui s'étend sur toute la largeur de la page ;
- une colonne gauche avec les liens de navigation ;
- une colonne centrale qui contient le contenu de la page ;

- une colonne droite qui comporte des liens mis en avant ou des mots-clés, ou encore des liens contextuels propres à la page ;
- une boîte pour le pied de page, qui contient les crédits du site, le lien de contact.

Le code général du site Écoute S'il Pleut peut être schématisé comme ceci :

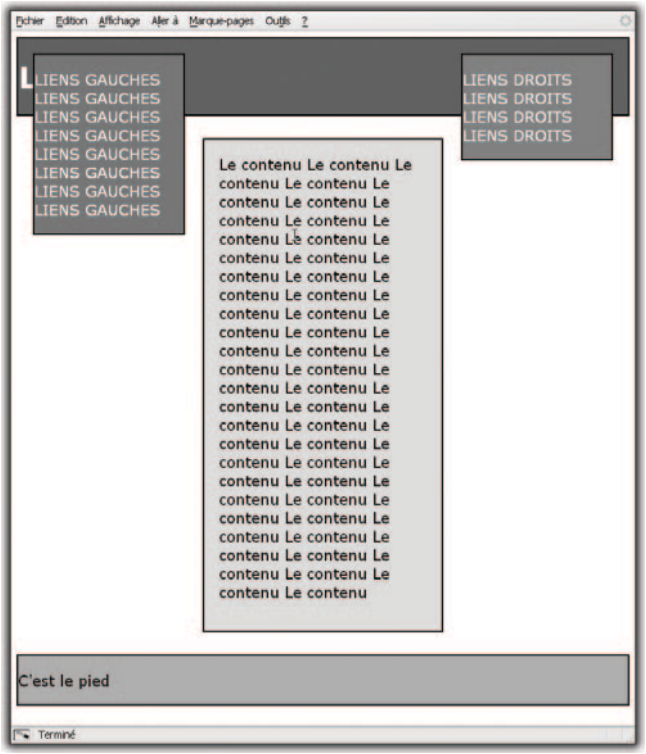
```
<div id="titre">LE TITRE </div>
<div id="contenu">
<div id="navigauche">
 LIENS GAUCHES<br \> … </div>
<div id="corps">
Le contenu…
 </div>
 <div id="navidroite">
LIENS DROITS<br \> … </div>
</div>
<div id="pied">
C'EST LE PIED
</div>
```

Les conteneurs se suivent verticalement dans l'ordre indiqué par le code source. Pour qu'on puisse les distinguer sur la page, ils vont être marqués par un fond de couleur. Maintenant, les boîtes sont positionnées avec une position absolue. Elles mordent sur la boîte `titre`.

```
body {
   font-family: Verdana, Arial, Helvetica, sans-serif;
   font-size: 1em;
}
div#titre {
   background-color: #BD4500;
   color: #FFF;
   border: 2px solid #000;
   }
div#corps {
   top: 0;
   padding: 0 0 0 10px;
   position: relative;
   margin: 30px 250px;
   padding: 0 20px 20px 20px;
   background-color: #E2EA8A;
   border: 2px solid #000;
}
div#navigauche {
```

```
width: 200px;
   left: 0;
   top: 0;
   position: absolute;
   margin: 30px;
   padding: 0px;
   color: #FFF;
   background-color: #5078CF;
   border: 2px solid #000;
}
div#navidroite {
width: 200px;
   position: absolute;
   right: 0;
   top: 0;
   margin: 30px;
   padding: 0px;
   color: #FFF;
   background-color: #E36503;
   border: 2px solid #000;
}
div#pied {
   left: auto;
   bottom: 20px;
   background-color: #7FE600;
   border: 2px solid #000;
   }
```

Fichier Edition Affichage Aller à Marque-pages Outils ?
LIENS GAUCHES
LIENS GAUCHES
LIENS GAUCHES
LIENS GAUCHES
LIENS GAUCHES
LIENS GAUCHES
LIENS GAUCHES
LIENS GAUCHES
LIENS DROITS
LIENS DROITS
LIENS DROITS
LIENS DROITS
Le contenu Le contenu Le contenu



▲ Figure 5.49 : *Absolus et relatifs avec modification des dimensions du navigateur*

Pour la boîte `titre`, il n'est pas nécessaire de préciser la position puisque cette boîte se place naturellement dans le flux, en haut de la page. Elle se distingue juste par sa couleur de fond.

L'ensemble des trois colonnes est formé des deux colonnes de navigation, qui sont en position absolue de chaque côté de la page, et de la colonne centrale qui, elle, est en position relative.

margin et padding

La propriété `margin` décrit les marges externes d'une boîte alors que `padding` décrit les marges internes.

Les propriétés `margin` et `padding` peuvent être suivies d'un tiret et d'une position (`top` : haut, `bottom` : bas, `left` : gauche, `right` : droite), comme `margin-bottom: 0` ou `padding-left: 20px`. Les valeurs peuvent être groupées sur une seule ligne comme ceci :

```
margin: 10px 20px 15px 10px;
```

- quatre valeurs : haut, droite, bas, gauche ;
- deux valeurs : haut et bas, puis droite et gauche ;
- une valeur : partout.

La colonne centrale est en position relative car elle se définit par rapport aux deux boîtes de navigation. Placer toutes les boîtes en position absolue est trop rigide et ne tient pas compte des différentes configurations des navigateurs. Par ailleurs, cela signifierait que le type d'affichage est défini de manière autoritaire et que le site, s'il est normalisé pour un affichage de 800×600 pixels, est complètement inadapté pour un affichage de $1\,280 \times 1\,024$ pixels. La place des boîtes de navigation est toujours à gauche et à droite et elles peuvent donc avoir une position absolue. Si la résolution de l'écran change, les boîtes de navigation resteront identiques ; par contre, la colonne centrale avec la position relative s'adaptera à la nouvelle largeur.

Si la marge n'est pas spécifiée dans la colonne centrale, la colonne de gauche passera sous la colonne centrale qui occupera toute la largeur. La colonne de droite, elle, passera au-dessus de la centrale. Ainsi, une partie du texte ne pourra être lue et les liens de navigation de la colonne de gauche ne seront pas accessibles.

La dimension z

Avec la propriété `z-index`, il est possible de faire passer une boîte au-dessus d'une autre. Pour cela, il suffit de doter la boîte d'un coefficient plus élevé pour qu'elle soit en haut de la pile. Le coefficient de `z-index` est 0 par défaut.

Pour que tout soit en place, il faut jouer sur la marge et sur le fait que les deux colonnes de navigation ont une largeur définie à 200 pixels. En mettant la marge horizontale à 250 pixels, le développeur du site laisse la place à chaque colonne latérale et ajoute une gouttière de 50 pixels.

Il subsiste un problème : il faut descendre les colonnes. La marge de 20 pixels n'est pas suffisante pour éviter de mordre sur la boîte de titre. C'est la propriété `top` qui permettra de descendre ces blocs. Ainsi en s'accroissant, le bloc descendra.

Après la manipulation qui permet de descendre les blocs, il subsiste un problème : ces deux blocs de navigation ne peuvent s'adapter à la hauteur de la boîte `titre`. La solution consiste à détourner un peu les règles CSS en insérant les trois colonnes dans une autre boîte, qui aura une position relative par rapport à la boîte `titre` qui est au-dessus. Cette boîte, appelée `contenu`, contient les blocs `navigauche`, `corps`, `navidroite`.

```
<div id="titre">LE TITRE </div>
 <div id="contenu">
<div id="navigauche">
 LIENS GAUCHES<br \>
 LIENS GAUCHES<br \>
 LIENS GAUCHES<br \>
 LIENS GAUCHES<br \>
 </div>
<div id="corps">
Le contenu Le contenu Le contenu Le contenu Le contenu
Le contenu Le contenu Le contenu Le contenu Le contenu
Le contenu
 </div>
 <div id="navidroite">
LIENS DROITS<br \>
LIENS DROITS<br \>
LIENS DROITS<br \>
LIENS DROITS<br \>
 </div>
 </div>
```

Dans les CSS, on ajoute les caractéristiques de la boîte `contenu`.

```
div#contenu{
height: 600px;
   position: relative;
   border: 2px solid #000;
   }
```

Ici les propriétés top des deux colonnes sont réglées sur la même valeur que la valeur de marge de la colonne du centre.

Une page à trois colonnes avec la position flottante

Après avoir étudié une forme de site à trois colonnes, avec les deux colonnes de navigation en position absolue et la colonne centrale en position relative, il est intéressant de se pencher sur un autre site à trois colonnes qui utilisent la position flottante (position: float;). Le site précédent obligeait à détourner le code CSS en utilisant une bidouille de conteneur, présent uniquement pour la bonne présentation. Pour les colonnes flottantes, le code est plus orthodoxe. Le code XHTML est le même, excepté que l'on enlève la boîte contenu.

```
<div id="titre">
 <h1>LE TITRE</h1>
</div>

 <div id="navigauche">
<p> LIENS GAUCHES<br \>…
</p>
 </div>
 <div id="navidroite">
 <p> LIENS DROITS<br \>
LIENS DROITS<br \>…
</p>
 </div>
 <div id="crops">
<p> Le contenu… </p>
</div>
<div id="pied">
 <p>C'est le pied </p>
</div>
```

La taille du texte de la navigation gauche a été allongée par rapport à la colonne de droite pour imiter une situation réelle. Le code CSS est le suivant :

```
div#titre {
 text-align: center;
 background-color: #CCCCCC;
 height: 60px;
 margin: 0px;
```

```
 padding: 1px;
}
div#navigauche {
 padding: 10px;
 width: 130px;
 float: left;
   background-color: #9999FF;
}
div#corps {
 padding: 10px;
 margin-left: 160px;
 margin-right: 160px;
   background-color: #5DC861;
}
div#navidroite {
 padding: 10px;
 width: 130px;
 float: right;
   background-color: #FFFF99;
}
div#pied {
 border-top: solid #000 1px;
 background-color: #CCCCCC;
 padding: 10px;
 text-align: center;
 clear: both;
}
```

Fichier Edition Affichage Aller à Marque-pages Outils ?
LE TITRE
LIENS GAUCHES
Le contenu Le contenu Le contenu

La position flottante permet à un bloc de flotter. C'est comme avec les images dans le HTML : le texte s'enroule naturellement autour. Le contenu aussi s'enroule autour de la boîte flottante, qui peut être à droite ou à gauche, comme les colonnes de navigation dans l'exemple, le bloc central s'insérant entre les deux, sans mordre sur les autres boîtes.

Pour la boîte `pied`, la propriété `clear` est utilisée. Cette propriété signifie au navigateur que de la place est libérée sur les côtés, à droite pour `clear: right`, à gauche pour `clear: left` et des deux côtés pour `clear: both`. Ainsi la boîte `pied` libère de la place de chaque côté et n'est donc pas collée à d'autres boîtes. La propriété de `clear` par défaut est `none`.

Plusieurs problèmes se posent, dont le titre qui mord un peu sur la limite avec Firefox. Cette anomalie n'en est pas vraiment une. En fait, le navigateur Firefox a une option pour afficher les caractères en grand et le titre mord donc sur le bord de la boîte. Pour rectifier, il suffit d'ajouter 10 pixels en hauteur.

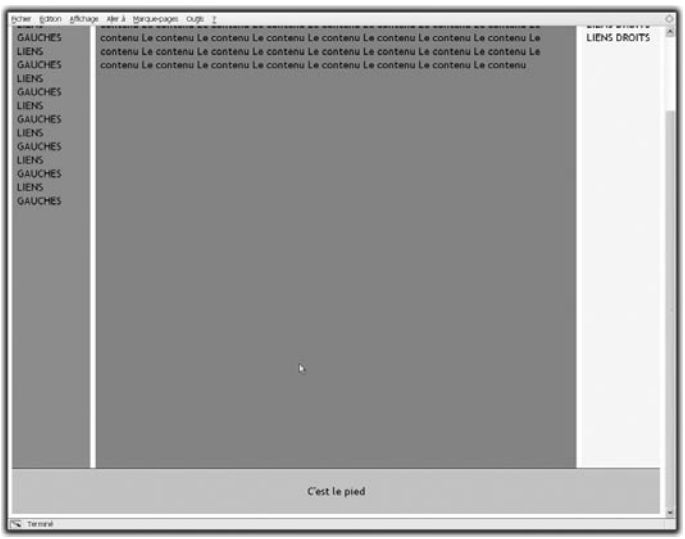
```
div#titre {
 text-align: center;
 background-color: #CCCCCC;
 height: 70px;
 margin: 0px;
 padding: 1px;
}
```

Une faute de goût affiche les gouttières en blanc entre les boîtes, ce qui tranche trop avec les couleurs d'arrière-plan des colonnes. Pour rectifier cela, le développeur peut ajouter une couleur de fond à toute la page, ce qui modifiera la couleur des gouttières.

```
body {
 margin: 0px;
 padding: 0px;
   background-color: #69F;
}
```

Le défaut le plus évident est le fait que les couleurs d'arrière-plan ne font qu'envelopper les textes et ne descendent pas jusqu'au bas de la page. Pour pallier cela, il faut ajouter la propriété `height: 100%;` dans chaque boîte.

```
div#navigauche {
 padding: 10px;
 width: 130px;
 float: left;
   background-color: #9999FF;
   height: 100%;
}
div#corps {
 padding: 10px;
 margin-left: 160px;
 margin-right: 160px;
   background-color: #5DC861;
   height: 100%;
}
div#navidroite {
 padding: 10px;
 width: 130px;
 float: right;
   background-color: #FFFF99;
   height: 100%;
}
```

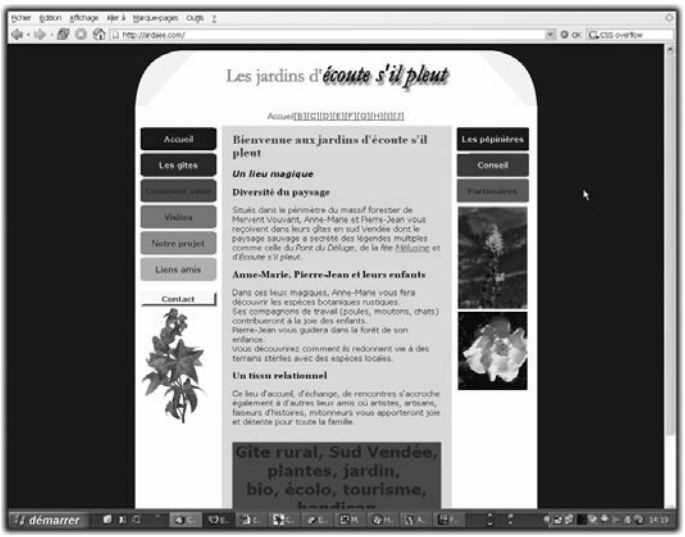


▲ Figure 5.51 : *Hauteur à 100 %*

Pour résoudre ces problèmes de positionnement et de compatibilité avec les navigateurs, le mieux est encore de faire un vrai site…

La construction du site Écoute S'il Pleut

Comme vu précédemment, le site Écoute S'il Pleut (`www.ardaee.com`) est à trois colonnes. C'est le moment où il faut prendre des options. Le choix va se porter sur un site qui est contenu dans un espace large de 770 pixels et qui est cependant assez souple pour permettre à l'utilisateur de modifier le format pour une meilleure lecture. Le choix se porte sur des boîtes flottantes plutôt que des boîtes en position absolue, qui posent le problème de lisibilité quand le visiteur modifie le format des polices utilisées. Dans ce cas, certaines boîtes peuvent en masquer d'autres et le site devient illisible. Cela arrive sur certains sites où le corps de police est minuscule, mais où il est possible de le modifier parce que la taille des polices n'est pas relative, ou lorsque les boîtes se télescopent et qu'une partie du texte est masquée. Cette erreur est rédhibitoire, même si le contenu est intéressant : le visiteur quitte le site et ne reviendra plus.



▲ Figure 5.52 : *Ardaee.com*

Le site utilisera aussi des astuces pour casser le schéma des boîtes rectangulaires alignées au cordeau, ce qui permettra de se confronter aux difficultés dues au fait qu'un certain nombre de navigateurs, Internet Explorer en particulier, ne respectent pas les recommandations du W3C en

matière de feuilles de style. Un navigateur qui n'applique pas les feuilles de style ne pose pas de problème car il affiche la page sans aucun style et celle-ci reste lisible. Par contre, un navigateur censé appliquer les feuilles de style, mais qui ne respecte pas les règles, est redoutable, d'autant plus s'il est largement utilisé. Il faudra donc faire du bricolage peu orthodoxe pour que l'affichage soit convenable sur les navigateurs les plus courants.

La page type du site présente un titre, une colonne gauche, un contenu, une colonne droite et une boîte de pied de page.

La construction commence par la création d'un conteneur qui épouse les limites de la balise `body`. Ce conteneur renferme tous les autres conteneurs et facilite la disposition de l'ensemble. Sa largeur est de 770 pixels, correspondant à l'affichage d'un écran à la résolution de 800×600 px, moins les marges occupées par le navigateur (glissière).

Cette surface permet de maîtriser les largeurs et surtout la distribution des boîtes et la disposition de l'ensemble. Cette boîte porte ici l'identifiant `contenu`. Dans la feuille de style, les premiers sélecteur et conteneur caractérisés sont le sélecteur `body` et le conteneur `contenu`.

```
body {
margin:0;
padding:0;
background:#233A00;
font:1em Verdana, Arial, Helvetica, sans-serif;
text-align: center;
}
div#contenu {
background:#ffffe1;
position: relative;
max-width: 770px;
margin: 10px auto 0 auto;
text-align: left;
padding: 0;
}
```

Les marges intérieure et extérieure sont remises à 0 pour effacer les réglages par défaut. Le `body` a une couleur de fond vert foncé et le sélecteur `contenu` une couleur de fond jaune très clair, tandis que le vrai contenu de la page (la colonne centrale) a une couleur qui tranche, jaune avec une pointe de vert. La propriété `text-align` est destinée à Internet Explorer : elle aligne les boîtes et met le conteneur `contenu` au milieu de

la page. Pour les autres navigateurs compatibles CSS, la façon d'aligner la boîte au centre est `margin: auto` dans la boîte elle-même. La position du conteneur `contenu` n'est pas précisée car il n'est pas nécessaire de la sortir du flux, puisqu'elle est la seule à son niveau. Comme il a fallu attribuer la valeur `center` à `text-align` à cause d'Internet Explorer, il faut rectifier pour les autres navigateurs avec `text-align: left` et remettre le texte avec un fer à gauche. De même, la propriété `max-width` n'est pas reconnue par Internet Explorer. Ainsi, il est utile d'ajouter un sélecteur uniquement compris par ce navigateur.

```
/* destiné à IE */
* html div#contenu {
width: 770px;
}
```

Sans ce sélecteur factice, le conteneur `contenu` remplirait toute la page dans toute sa largeur. Les autres boîtes se placent dans cette boîte. Le titre est placé dans le flux. Les trois colonnes sont flottantes à gauche.

```
#titre {
text-align: center;
margin: 15px;
clear: both;
}
```

L'indication `clear: both` est une sécurité. En effet, les boîtes qui suivent le titre sont flottantes, pour s'assurer qu'il occupe toute la largeur sans qu'aucune boîte ne s'y glisse. Cette propriété n'est pas nécessaire ici, mais elle peut être utile dans d'autres circonstances.

Il est important de s'engager dans un processus qui aboutit à une solution ou à une autre car l'application des CSS demande de la patience et de la créativité. Il ne suffit pas d'appliquer les règles qui sont théoriques, d'autant qu'elles ne s'appliquent pas de la même façon selon les navigateurs. Cette conformité est ce qui est le plus consommateur de temps avec les CSS. Au fil des versions des navigateurs, on peut espérer une certaine normalisation pour que les développeurs puissent se concentrer sur l'aspect de la page plutôt que sur du code bricolé. Les boîtes flottantes sont intéressantes parce qu'elles permettent, de la meilleure façon qui soit, de placer des boîtes côte à côte. Cependant, ce processus est fragile et il suffit parfois d'un ajout pour bousculer le bel ordonnan-

cement (et il n'est pas forcément aisé d'en deviner la cause). Le travail sur les CSS demande une certaine connaissance du code, et du pragmatisme doublé d'opportunisme.

Voici l'hypothèse de départ :

- La largeur du contenu est définie à 770 px, quel que soit l'affichage de l'écran.
- La largeur de la colonne centrale est définie à 400 px.
- Il reste 330 px pour les deux colonnes latérales de navigation et leurs marges externes.

Voici le raisonnement suivi :

- Ici, il est impossible de mettre une boîte en position absolue car la page est centrée et la distance jusqu'au bord du navigateur est inconnue car elle dépend de la définition de l'écran.
- En mettant la colonne centrale en position relative et en utilisant des colonnes latérales flottantes, on se heurte à une difficulté : la colonne de droite est renvoyée en dessous des deux autres colonnes.
- Faisons flotter les trois colonnes et observons le résultat dans un navigateur conforme comme Firefox 1.5 :
  - La colonne gauche flottant à gauche, la colonne centrale flottant à droite, la colonne droite flottant à droite. Résultat : la colonne centrale va à droite et la colonne de droite est renvoyée en dessous.
  - La colonne gauche flottant à gauche, la colonne centrale flottant à gauche, la colonne droite flottant à droite. Résultat : tout se tient mais si l'on fait un zoom arrière sur la page, la colonne de droite fuit sur la droite, ce qui crée une dysharmonie.
  - La colonne gauche flottant à gauche, la colonne centrale flottant à gauche, la colonne droite flottant à gauche. Résultat : tout se tient.

Par conséquent, la meilleure solution est la troisième. Il reste cependant plusieurs écueils : il va falloir ruser pour faire accepter cette mixture de code à Internet Explorer. Il faudra donc ajouter des ingrédients et lui faire avaler le tout. En fait, le navigateur conforme sert d'étalon, il faut ensuite ajuster le résultat pour plaire au plus grand nombre possible de navigateurs.

```
div#corps {
float: left;
   background-color: #E2EA8A;
   font-size: 0.75em;
 width: 400px;

padding: 0 20px 20px 20px;
background-color:transparent;
background-repeat: repeat-y;
background-position:top;
background-image:url("images/fond_corps.gif");
}

div#navigauche {
top: 100px;
   z-index: 2;
   margin: 0 10px;
   padding: 0px;
   float: left;
   color: #E2EA8A;
   font-size: 0.75em;
}
div#navidroite {
   margin: 0 10px;
   top: 100px;
   right: 0;
   padding: 0px;
float: left;
   color: #E2EA8A;
   font-size: 0.75em;
clear: right;
}
```

Sur le navigateur Firefox, la page est parfaitement ordonnancée. Il reste à lui adjoindre quelques courbes pour adoucir les angles droits tellement impersonnels. Pour cela, on peut utiliser toutes sortes de logiciels graphiques, vectoriels ou non, mais le plus facile pour avoir un résultat propre est de recourir à un logiciel vectoriel comme Illustrator. Ensuite, il n'y aura plus qu'à insérer ce motif dans la boîte `contenu` comme motif de fond, en le plaçant en haut et sans répétition.

Au moment de l'exécution, tout n'est pas parfaitement aligné. Il faut alors ajouter une bordure de la même couleur que celle du fond de page, pour couper ce qui dépasse et qui n'est pas droit.

```
div#contenu {
background:#ffffe1;
background-image:url("images/haut.gif");
background-repeat:no-repeat;
background-position:top left;
max-width: 770px;
margin: 10px auto 0 auto;
text-align: left;
border: 1px solid #233A00;
padding: 0;
}
```

Il faut maintenant s'occuper des internautes qui utilisent Internet Explorer en adjoignant au code des "verrues" qui seront comprises uniquement par ce navigateur, afin d'assurer un aspect uniforme pour tous.

Une fois le problème de la largeur du conteneur `contenu` résolu, il reste le problème de la largeur du conteneur `corps`. On ajoute un sélecteur qui ne sera pris en compte que par Internet Explorer. Comme Internet Explorer mesure les boîtes de manière particulière, on en tient compte en mettant les marges en négatif.

```
* html div#corps {
margin: 0 -40px 20px -40px;
padding: 0 40px 20px 40px;
}
```

Le processus doit être complété avec un bidouillage dans le sélecteur `corps`, qui réajuste l'affichage dans Firefox.

```
div#corps {
float: left;
   background-color: #E2EA8A;
   font-size: 0.75em;
 voice-family: "\"}\"";
 voice-family: inherit;
/* On annule l'information précédente */
    width: 400px;
   margin: 0 0 20px 0;
   padding: 0 20px 20px 20px;
background-color:transparent;
background-repeat: repeat-y;
background-position:top;
background-image:url("images/fond_corps.gif");
}
```

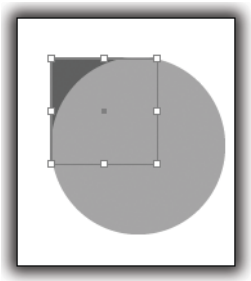
La création des coins ou des chapeaux de boîte

La création des coins peut paraître un sujet qui ne concerne pas les CSS. Mais c'est une technique qui mérite d'être traitée dans ce livre car tout développeur est obligé de s'en servir de temps en temps. La meilleure façon d'opérer est de recourir à un logiciel de type vectoriel comme Illustrator ou le logiciel libre Sodipodi téléchargeable sur `www.framasoft.net/article1589.html`.

▲ Figure 5.53 : *Framasoft*

Le processus consiste à créer un quart de cercle dans un carré ou un rectangle. Il faut commencer par inscrire le quart du cercle dans le carré. Le cercle est de la couleur de la partie intérieure de la boîte et le carré de la partie extérieure. Avec l'effet miroir, on copie le motif en miroir puis on suspend les deux objets à une bordure de 770 px pour le site Écoute S'il Pleut. Il faut se servir de l'outil d'alignement pour que les deux coïncident parfaitement.

Ensuite l'exercice consiste à enlever les points d'ancrage du cercle de façon à avoir un arc de cercle à angle droit qui fait le quart du cercle, parfaitement superposé sur l'angle droit du carré. Cette manière un peu rudimentaire permet de moduler l'arc de cercle à son gré.



◄ Figure 5.54 : *Avec un logiciel vectoriel*

Sinon, la manière de faire la plus simple pour avoir une tête et un pied de boîte est de créer un rectangle à coins arrondis sur un fond transparent puis de le couper pour en extraire la tête et le pied.

◄ Figure 5.55 : *Avec un rectangle à bords arrondis*

La couleur transparente

Internet Explorer a un problème avec la transparence, surtout pour les images PNG, dont la transparence est colorée en bleu clair.

Les difficultés rencontrées

Les difficultés rencontrées tiennent aux présupposés décidés au départ. Le fait de définir la largeur de la boîte `contenu` à 770 pixels permet d'avoir un site avec des arrondis, visible par tous les navigateurs. Cette force est aussi une fragilité. En effet, la largeur de la partie centrale est difficile à maîtriser sous Internet Explorer, ce qui fait que cette colonne mord sur les deux autres sises de chaque côté. Il est aussi difficile d'ajouter des images dans les deux colonnes latérales sans bousculer tout le reste. Il faut en fait calculer précisément la largeur des images à insérer. Le tout n'est pas précisément au centre car on n'a pas encore les éléments flottants centrés, mais tout se tient quand même.

▲ Figure 5.56 : *Réduction de la fenêtre avec la version 1*

▲ Figure 5.57 : *Réduction de la fenêtre avec la version 1*

▲ Figure 5.58 : *Version 1 sous IE*

▲ Figure 5.59 : *Version 1 avec fenêtre réduite sous IE*

Finalement, pour le site `ardaee.com`, il a fallu revenir aux boîtes à positions relatives et absolues car les boîtes flottantes pour les trois colonnes centrées dans le navigateur n'ont pas de gouttières précises. Les

CSS obligent à des compromis pour la disposition. Les boîtes principales flottantes sont plus adéquates et simples pour une page alignée à gauche de la fenêtre du navigateur.

```
div#contenu {
   background: #ffffe1 url("images/haut_droite_petit.gif")
no-repeat right top;
   width: 85%;
   margin: 10px auto 0 auto;
   text-align: left;
   border: 1px solid #90C528;
   padding: 0;
}
.gauche{
   padding-top : 1em !important;
   background: transparent url("images/haut_gauche_petit.gif")
   ✂ no-repeat left top;
}

/* destiné à IE */
* html div#contenu {
   text-align: center;
   width: 770px;
}

div#corps {
   position: relative;
   background-color: #E1EB89;
   font-size: 0.75em;
    voice-family: "\"}\"";
    voice-family: inherit;
/* On annule l'information précédente */
   width: 500px;
   margin: 0 auto 20px auto;
   padding: 0 20px 20px 20px;
   background-color:transparent;
   background-repeat: repeat-y;
   background-position:top;
   background-image:url("images/fond_corps.gif");
}

* html div#corps {
   word-wrap: break-word;
   text-align: left;
   z-index: 1;
   width: 300px;
   margin: 0 20px 20px 20px;
   padding: 0 60px 20px 60px;
   position: relative;
```

```
}

div#navigauche {
   background: #FFFFE1;
   width: 200px;
   left: 0;
   top: 40px;
   position: absolute;
   margin: 10%;
   padding: 0px;
   color: #FFF;
   font-size: 0.75em;
}

div#navidroite {
   background: #FFFFE1;
   width: 200px;
   position: absolute;
   right: 0;
   top: 40px;
   margin: 10%;
   padding: 0px;
   color: #FFF;
   background-color: #FFFFE1;;
   font-size: 0.75em;
}

* html div#navigauche {
   background: #ffffe1 url("images/haut_gauche_petit.gif")
no-repeat left top;
   top: 30px ;
   left: 40px ;
   text-align: center;
   padding: 20px 20px 20px 20px;
}

* html div#navidroite {
   background: #ffffe1 url("images/haut_droite_petit.gif")
no-repeat right top;
   right: 40px ;
   top: 30px ;
   text-align: center;
   padding: 20px 20px 20px 20px;
}
```

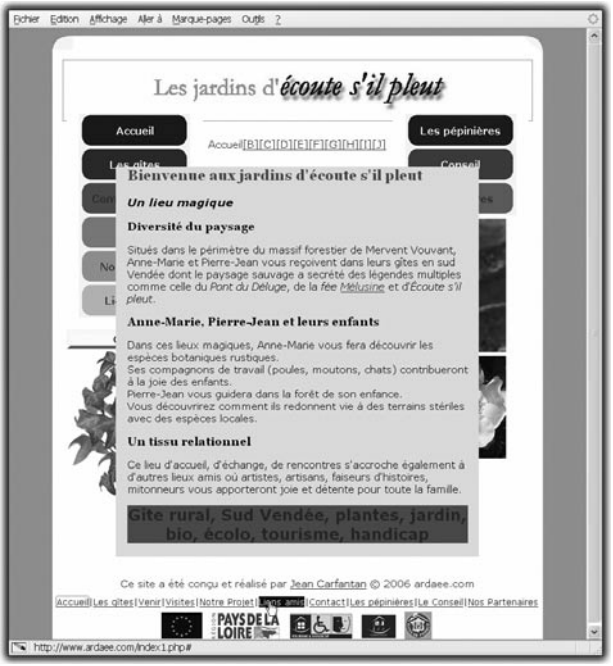
Les coins ont été réduits à 30 pixels et une balise `hr` a été ajoutée pour, avec le conteneur `contenu`, apposer les deux coins supérieurs.

▲ Figure 5.60 : *Le nouveau site*

```
<div id="contenu">
<div class="gauche"></div>
…
<div>
```

Les deux menus latéraux se glissent sous la colonne centrale dans une fenêtre réduite, ce qui oblige à tripler la barre de navigation en pied de page afin que les visiteurs qui disposent d'écran de faible résolution, puissent naviguer.

▲ Figure 5.61 : *Les liens en pied de page*



▲ Figure 5.62 : *La version 2 avec réduction de fenêtre*

▲ Figure 5.63 : *Version 2 sous IE*

▲ Figure 5.64 : *Version 2 avec fenêtre réduite sous IE*

Sans doute pour rendre le site "parfait", faudrait-il passer plus de temps et complexifier le code, mais le but est avant tout de communiquer et de se faciliter la tâche avec les CSS. Il est clair que les CSS n'ont pas encore atteint leur maturité mais elles restent néanmoins un progrès par rapport aux tables.

6

Les règles d'accession à la propriété

Internet mûrit et s'intéresse à l'accessibilité, c'est-à-dire qu'il se préoccupe de savoir si tout le monde a accès au contenu des pages. Internet, et l'informatique en général, s'est construit autour d'un utilisateur sans handicap. Mais qu'en est-il par exemple des personnes qui ont une mauvaise vue, voire des aveugles, des daltoniens, des personnes qui ne peuvent utiliser une souris ? Un certain nombre de règles permettent aux handicapés, quel que soit leur niveau d'handicap, de naviguer sur le Web et de profiter de toutes les ressources disponibles. En fait, conformer un site aux critères de l'accessibilité oblige à porter une attention particulière à la communication, à la rigueur et à la cohérence, ce qui profite aux handicapés, et plus généralement à tous, y compris au webmaster. Quiconque peut, un jour, se retrouver invalide temporairement ou définitivement, au point de ne plus pouvoir utiliser Internet.

◄ Figure 6.1 : *Accessibilité*

Définition de l'accessibilité

L'accessibilité a été définie comme le fait de mettre le Web et ses services à la disposition de tous les individus, quels que soient leur matériel ou logiciel, leur infrastructure réseau, leur langue maternelle, leur culture, leur localisation géographique ou leurs aptitudes physiques et mentales.

L'accessibilité est une philosophie de vie. C'est la vraie démocratie, quand la majorité porte son attention sur les problèmes de la minorité. L'accessibilité commence quand un développeur, lui-même utilisateur de PC, s'inquiète du visiteur qui possède un Macintosh ou de celui qui possède un vieux PC avec un affichage médiocre. Dans des pays comme la France, où la polarité est très forte, il est difficile de s'accorder sur des mesures qui se penchent sur la condition de vie d'un certain nombre de concitoyens. Et pourtant une loi a été votée dans ce sens.

La France a voté une loi en février 2005 dont un article impose que les sites web de l'État et des collectivités territoriales répondent aux normes internationales d'accessibilité. Peu de sites respectent encore ces règles,

mais cette norme deviendra incontournable et s'inscrira de plus en plus dans le cahier des charges de création de sites. Si ce label aide à l'accession de tous au Web, il rehausse, en même temps, l'image de marque du site et donc de la société qu'il représente. La qualité des sites accessibles est grandement éprouvée car l'accessibilité oblige à se poser des questions sur l'ergonomie, la communication, etc. Ce type de site offre un panel complet des moyens de communication, avec une diversité de passerelles.

Dans la vie réelle, l'humain a accès à de nombreux stimuli (visuels, auditifs, olfactifs…), qui lui permettent de communiquer avec le monde. L'accessibilité restituer cette polyphonie de la vie sur le Web. Les technologies se développent de jour en jour pour rendre la navigation sur le Web plus agréable pour tous. Ainsi le site Agoravox propose une fonctionnalité qui génère un fichier MP3 où une voix lit les articles du site. Agoravox (`www.agoravox.fr`) utilise, pour ce faire, la technologie Readspeaker, qui oblige à davantage de rigueur car un texte mal écrit est lu plus difficilement et une simple erreur de ponctuation nuit à la compréhension d'une phrase. Les rédacteurs auront tout intérêt à passer leurs textes à la moulinette de Readspeaker : ce faisant, ils les amélioreront notablement.

▲ Figure 6.2 : *Agoravox*

La rigueur, l'ergonomie, la structuration des contenus sont des éléments clés pour que ces contenus s'adaptent facilement à n'importe quel média et se transportent avec souplesse d'un site à l'autre avec un minimum de traitement. Désormais, l'internaute pourra choisir son mode de connaissance d'une page web : il préférera l'entendre, ou naviguer par la voix, le clavier, la voir sur son téléphone ou sa télévision. C'est cette diversité qui répond le mieux à l'évolution générale de la société. Avant d'aller plus loin, validez vos pages sur `www.ocawa.com`.

▲ Figure 6.3 : *Ocawa*

6.1 Le doctype

Le `doctype` donne la grammaire selon laquelle la page est construite. Il parle directement au navigateur. Si les pages sont déjà construites sans `doctype`, il suffit d'ajouter une déclaration de type :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//FR"
✂< "http://www.w3.org/TR/html4/loose.dtd">
```

Si la présentation de la page est modifiée, il suffit d'ajouter :

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//FR">
```

6.2 La langue

Il est important, pour les navigateurs qui naviguent en lisant le contenu des pages pour les malvoyants ou les aveugles, de connaître la langue utilisée dans ces pages. Cette indication se fait dans la balise `html` et a une syntaxe particulière selon le `doctype`.

◄ Figure 6.4 : *Langues*

Avec le HTML 4, la balise `html` est modifiée par l'ajout de la propriété `lang` avec comme valeur le code ISO.

```
<html lang="fr">
```

Pour le XHTML 1.0, la balise `html` est enrichie de deux propriétés. La propriété `lang` subsiste mais elle n'est présente que pour assurer la compatibilité avec les anciens navigateurs.

```
<html xmlns="http://www.w3.org/1999/xhtml"
lang="fr" xml:lang="fr">
```

Enfin, le XHTML 1.1 exige la propriété `xml:lang` dans la balise `html`.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
```

Si une page est rédigée en plusieurs langues, la propriété `xml:lang` peut enrichir une balise `p` par exemple, ou d'autres balises, pour spécifier les langues en question.

6.3 Les titres

Le titre est un élément primordial dans une page. Il s'affiche sur la bordure supérieure du navigateur et annonce, comme une enseigne de magasin, ce qui sera mis à la disposition de l'internaute, au niveau du site et de la page elle-même. Il s'agit d'inciter le visiteur à cliquer sur le lien du titre. Il est recommandé d'inclure le nom du site dans tous les titres des pages.

▲ Figure 6.5 : *Titre*

6.4 Les liens

Cliquer sur un lien est une aventure, surtout pour l'internaute qui ne bénéficie pas du haut débit. Le lien se doit d'être court et n'a pas forcément à expliciter sa destination. C'est là qu'intervient la propriété `title`, qui affiche une étiquette explicative dans le navigateur. L'intitulé dans un lien ou dans un `title` ne doivent pas dépasser 80 caractères.

▲ Figure 6.6 : *Title*

6.5 Les éléments de formulaires

Pour l'accessibilité d'un formulaire, il est judicieux d'utiliser la balise `fieldset`, qui encadre les champs regroupés par signification. Dans un questionnaire, il est ainsi possible de regrouper dans un `fieldset` tous les champs concernant l'état civil, puis dans un autre ceux définissant les goûts personnels, enfin dans un troisième ceux relatifs aux observations personnelles. Un `fieldset` ne touche pas à la présentation directement, mais le navigateur trace un encadrement avec des coins arrondis.

```
<form action="…" method="post">
 <p>
 <fieldset>
  <legend>État civil</legend>
  <label for="nom">Nom :</label>
<input id="nom" type="text" tabindex="1">
  <label for="prenom">Prênom :</label>
<input id="prenom" type="text" tabindex="2">
  <label for="adresse">Adresse :</label>
<input id="adresse" type="text" tabindex="3">
</fieldset>
</p>
 <fieldset>
  <legend>Questionnaire</legend>
Ce site est :<br />
  <input name="avis"
         type="checkbox"
         value="ergonomique" id="ergonomique" tabindex="20">
<label for="ergonomique"> Ergonomique </label>
<input name="avis"
         type="checkbox"
         value="beau" id="beau" tabindex="21">
<label for="beau"> Beau </label>
<input name="avis"
         type="checkbox"
         value="interessant" id="interessant" tabindex="22">
<label for="interessant"> Intéressant </label>
<input name="avis"
         type="checkbox"
         value="complet" id="complet" tabindex="23">
<label for="complet"> Complet </label>
</fieldset>
</p>
<p>
 <fieldset>
  <legend>Améliorations</legend>
  Pensez-vous que nous puissions l'améliorer ?
```

```
  <input name="amelioration"
         type="radio"
         value="oui" id="affirmative" tabindex="35">
<label for="affirmative"> Oui </label>
  <input name="amelioration"
         type="radio" id="negative"
         value="non" tabindex="35"><label for="negative"> Non
</label></p>
  <p><label for="suggestions">Si vous pensez que nous
pouvons améliorer ce site, faites nous part de vos
suggestions :</label>
  <textarea name="suggestions" id="suggestions"
            rows="20" cols="50" tabindex="40">
  </textarea>
 </fieldset>
 </p>
</form>
```

La propriété `tabindex` permet de naviguer d'un champ à l'autre. L'ordre va du plus petit au plus grand sans qu'il soit nécessaire d'avoir des valeurs qui se suivent. Un champ avec une valeur négative sera ignoré.

Dans l'exemple, la balise `label` est utilisée pour accompagner les champs du formulaire. Cette balise est reliée par sa propriété `for` à la propriété `id` du champ de formulaire.

▲ Figure 6.7 : *Fieldset*

6.6 Les tableaux

La plupart du temps, les tableaux ne sont pas utilisés pour ordonner des données, comme c'était le cas à l'origine. On détourne leur fonction pour structurer une page et disposer les différents éléments, ce qui pose problème.

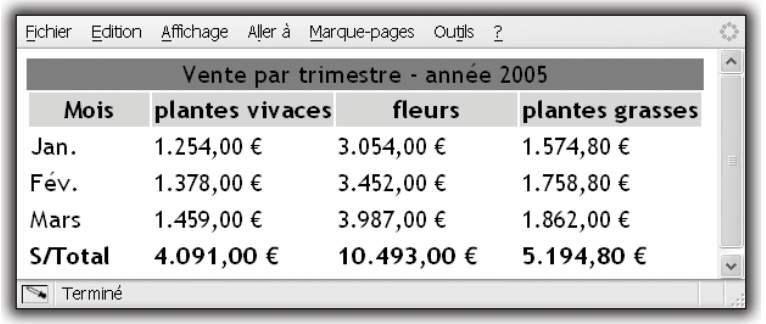
Un tableau est habituellement créé comme ceci :

```
<TABLE BORDER>
          <CAPTION>Titre du tableau</CAPTION>
          <TR>
             <TH>1re colonne</TH>
             <TH>2e colonne</TH>
             <TH>3e colonne</TH>
          </TR>
          <TR>
             <TD>A</TD>
             <TD>B</TD>
             <TD>C</TD>
          </TR>
          <TR>
             <TD>D</TD>
             <TD>E</TD>
             <TD>F</TD>
          </TR>
        </TABLE>
```

La balise `caption` donne le titre du tableau juste au-dessus de celui-ci. La balise `th` donne le titre des colonnes. On peut ajouter la propriété `summary`, qui ne sera pas affichée mais qui sera perçue par les navigateurs audio.

```
<table border="0" summary="Les 6 premières lettres de l'alphabet. Le
➥< tableau est constitué de 2 rangs et 3 colonnes.">
```

La norme d'accessibilité recommande d'utiliser les propriétés `headers`, et la propriété `abbr` dès que le titre d'une colonne contient plus de 15 caractères (voir Figure 6.8).



▲ Figure 6.8 : *Tableau des ventes*

```
<table summary="Tableau des ventes des plantes au cours
du premier trimestre de l'année 2005">
<caption>Vente par trimestre - année 2005</caption>
<colgroup align="char" char=",">
<col width="2*" />
<col width="3*" />
<col width="3*" />
<col width="3*" />
</colgroup>
<thead>
<tr>
<th id="entete1">Mois</th>
<th id="entete2" abbr="vivaces">plantes vivaces</th>
<th id="entete3">fleurs</th>
<th id="entete4" abbr="grasses">plantes grasses</th>
</tr>
</thead>
<tfoot>
<tr>
<td headers="entete1">S/Total</td>
<td headers="entete2">4.091,00 ¤</td>
<td headers="entete3">10.493,00 ¤</td>
<td headers="entete4">5.194,80 ¤</td>
</tr>
</tfoot>
<tbody>
<tr>
<td headers="entete1">Jan.</td>
<td headers="entete2">1.254,00 ¤</td>
<td headers="entete3">3.054,00 ¤</td>
<td headers="entete4">1.574,80 ¤</td>
</tr>
<tr>
```

```
<td headers="entete1">Fév.</td>
<td headers="entete2">1.378,00 ¤</td>
<td headers="entete3">3.452,00 ¤</td>
<td headers="entete4">1.758,80 ¤</td>
</tr>
<tr>
<td headers="entete1">Mars</td>
<td headers="entete2">1.459,00 ¤</td>
<td headers="entete3">3.987,00 ¤</td>
<td headers="entete4">1.862,00 ¤</td>
</tr>
</tbody>
</table>
```

La balise `colgroup` envoie au navigateur des informations sur le nombre et la proportion des colonnes et permet au tableau de s'afficher progressivement. Elle doit être placée juste après la balise `table`.

La propriété `width` donne un ordre de grandeur de la colonne par rapport à la taille totale du tableau. Grâce à ce type de positionnement, le tableau se déploie harmonieusement sur la page.

Les propriétés `align="char"` et `char=","` gèrent l'alignement dans la colonne en question. Elles demandent ici au navigateur d'aligner la colonne sur le caractère virgule.

Trois groupes sont ici déployés, matérialisés par les balises `thead`, `tfoot` et `tbody`. La balise `thead` regroupe toutes les balises `th` et définit l'en-tête du tableau. Elle est suivie par la balise `tfoot`, qui circonscrit le pied du tableau. Enfin vient le groupe `tbody`, qui caractérise les cellules du corps du tableau. La feuille de style est simple.

```
caption {
border: thick;
background-color: #00CC36;
}
#entete1, #entete2, #entete3, #entete4 {
border: thick;
background-color: #F6E730;
}
tfoot {
font-weight: bold;
}
```

6.7 Les listes

Les énumérations de liens ou d'autres items sous forme de listes doivent pouvoir être identifiées comme telles. Une énumération est un ensemble, et non pas plusieurs éléments isolés les uns à côté des autres.

Un exemple courant de liste est celle utilisée pour afficher des liens, y compris dans un menu de navigation.

Voici les liens vers les journaux français :

```
<img src="/images/puce_sympa.gif" width="8" height="8">
<a href="http://www.lemonde.fr/">Le Monde</a> <br>
<img src="/images/puce_sympa.gif" width="8" height="8">
<a href="http://www.liberation.fr/">Libération</a> <br>
<img src="/images/puce_sympa.gif" width="8" height="8">
<a href="http://www.lefigaro.fr/">Le Figaro</a> <br>
<img src="/images/puce_sympa.gif" width="8" height="8">
<a href=" http://www.humanite.presse.fr/">L'Humanité</a> <br>
<img src="/images/puce_sympa.gif" width="8" height="8">
<a href=" http://www.la-croix.com/">La Croix</a> <br>
```

◄ Figure 6.9 : *Journaux français 1*

Rien, à part la vision de la page, ne permet au navigateur de savoir qu'il s'agit d'une liste. Cela a l'apparence d'une liste, mais sans en être une. Une solution pour que le navigateur s'y retrouve est de lui fournir des indices. Il s'agit d'insérer les propriétés `alt` et `title` avec respectivement comme valeurs * et une chaîne vide.

```
<img alt="*" title="" src="/images/puce_sympa.gif" width="8"
height="8">
<a href="http://www.lemonde.fr/">Le Monde</a> <br>
<img alt="*" title="" src="/images/puce_sympa.gif" width="8"
height="8">
<a href="http://www.liberation.fr/">Libération</a> <br>
<img alt="*" title="" src="/images/puce_sympa.gif" width="8"
height="8">
<a href="http://www.lefigaro.fr/">Le Figaro</a> <br>
<img alt="*" title="" src="/images/puce_sympa.gif" width="8"
height="8">
<a href=" http://www.humanite.presse.fr/">L'Humanité</a> <br>
<img alt="*" title="" src="/images/puce_sympa.gif" width="8"
height="8">
<a href=" http://www.la-croix.com/">La Croix</a> <br>
```

◄ Figure 6.10 : *Journaux français 2*

Cette solution n'est cependant pas optimale car elle ne fait pas intervenir les feuilles de style. Une technique plus appropriée consiste à introduire la propriété `list-style` dans un sélecteur `ul`.

```
ul.journauxl {
list-style: url(/images/puce_sympa.gif) disc;
}
```

La liste devient alors :

```
<ul class="journaux">
<li><a href="http://www.lemonde.fr/">Le Monde</a></li>
<li><a href="http://www.liberation.fr/">Libération</a></li>
<li><a href="http://www.lefigaro.fr/">Le Figaro</a></li>
<li><a href=" http://www.humanite.presse.fr/">
L'Humanité</a> </li>
<li><a href=" http://www.la-croix.com/">La Croix</a></li>
</ul>
```

L'image sera affichée comme une puce par la plupart des navigateurs. Les autres afficheront des puces à la place de l'image.

◄ Figure 6.11 : *Journaux français 3*

Certains ne se posent pas de questions et se contentent d'aligner des liens dans une boîte.

```
<div align="left">
 <a href="http://www.lemonde.fr/">Le Monde</a><br />
 <a href="http://www.liberation.fr/">Libération</a><br />
 <a href="http://www.lefigaro.fr/">Le Figaro</a><br />
 <a href=" http://www.humanite.presse.fr/">L'Humanité</a>
<br />
 <a href=" http://www.la-croix.com/">La Croix</a>
</div>
```

Ils pourraient faire de même en désignant clairement cette énumération comme une liste.

```
ul.journaux {
list-style: none;
}
```

Quelle que soit la manière, le balisage de la liste n'a pas changé, comme dans l'exemple précédent :

```
<ul class="journaux">
<li><a href="http://www.lemonde.fr/">Le Monde</a></li>
<li><a href="http://www.liberation.fr/">Libération</a></li>
```

```
<li><a href="http://www.lefigaro.fr/">Le Figaro</a></li>
<li><a href=" http://www.humanite.presse.fr/">L'Humanité</a>
</li>
<li><a href=" http://www.la-croix.com/">La Croix</a></li>
</ul>
```

◄ Figure 6.12 : *Journaux français 4*

Quelquefois, il vaut mieux ne pas aller trop vite et réfléchir quelques secondes pour éviter des désagréments aux visiteurs du site. Quand l'habitude est prise, le développeur n'a même plus besoin d'y penser, cela devient un réflexe.

6.8 Les images

La propriété `alt` est toujours présente et a une valeur, sauf pour les éléments sans signification, qui ne sont là que pour ajuster la mise en page, comme les pixels transparents. Pour les éléments graphiques non signifiants, la balise `alt` doit être vide, sans texte ni espace. Ainsi, les navigateurs qui lisent le contenu des pages ne tiendront pas compte de ces éléments. La balise `alt` permet de naviguer sans afficher les images pour les navigateurs texte comme Lynx (jeu de mots construit sur le terme anglais "links") `http://lynx.isc.org/lynx2.8.5`. Un émulateur Lynx permet d'accéder à l'affichage de Lynx sans qu'il soit nécessaire de le télécharger (`www.delorie.com/web/lynxview.html`).

Les pixels transparents

Une des astuces utilisées par les développeurs et les graphistes consiste à ajouter des images GIF transparentes pour caler les colonnes et les différents éléments de la page.

Le nombre de caractères dans la propriété `alt` ne peut dépasser 60, excepté quand l'image sert de lien, la limite étant alors de 80 caractères.

En contradiction avec le XHTML

Une balise `alt` sans valeur utilisée pour les éléments graphiques décoratifs ou utilitaires est en contradiction avec l'impératif du XHTML de donner une valeur à toutes les propriétés. C'est un choix. Entre l'accessibilité et les règles strictes du XHML, il faut parfois choisir. Il reste cependant possible de lui donner comme valeur un astérisque.

6.9 Les abréviations

Les balises `acronym` et `abbr` ont la propriété `title`, qui donne aux navigateurs la signification complète des abréviations ou autres acronymes.

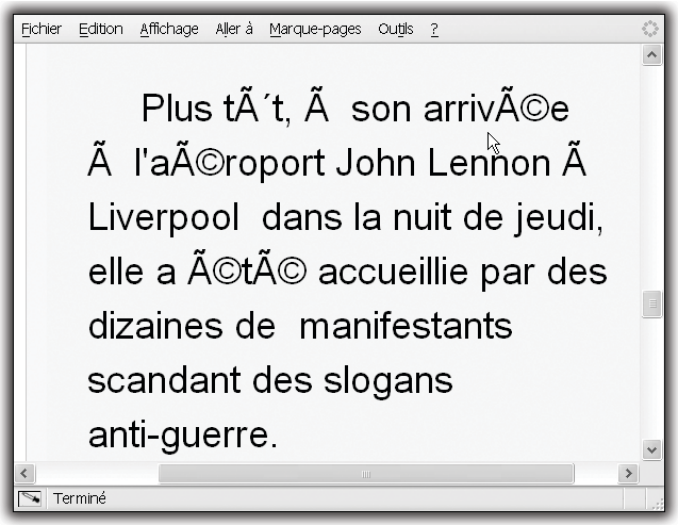
```
Avec la <acronym title="Société Nationale des Chemins de fer
➥< Français">SNCF</acronym>, c'est possible, <abbr title=
➥< "Monsieur">M</abbr> Dupont!
```

C'est particulièrement intéressant pour les navigateurs qui lisent les pages web pour les non-voyants.

6.10 Le charset ou tableau de caractères

La balise `meta` a obligatoirement comme propriété `charset`. Le `charset` sert à indiquer au navigateur le codage des caractères. Le codage par défaut sur Internet est ISO-8859-1.

```
<meta http-equiv="Content-type" content="text/html;
✂ charset=iso-8859-1" />
```



▲ Figure 6.13 : *Page avec un charset 8859-1 et des caractères UTF-8*

Le problème du système ISO-8859-1 est qu'il lui manque trois caractères en français : Œ, œ et Ÿ. Pour rectifier le problème, la norme ISO a ajouté le codage ISO-8859-15.

```
<meta http-equiv="Content-type" content="text/html;
✂ charset=iso-8859-15" />
```

Il existe aussi l'UTF-8, qui serait l'idéal et qui est la norme XML mais en HTML. L'affichage correct n'est pas certain.

```
<meta http-equiv="Content-type" content="text/html; charset=utf-8" />
```

6.11 La navigation

La navigation paraît aller de soi pour tout un chacun. Les liens sont affichés dans la page à un endroit visible, la plupart de temps en haut, à gauche ou encore à droite de la page. D'un point de vue visuel,

l'internaute ne peut pas les manquer. Mais pour un non-voyant dont le navigateur lit la page de manière linéaire (ce qui est logique visuellement), cela peut devenir un cauchemar quand il faut passer par des contenus répétitifs et fastidieux. Certains aménagements simples sont nécessaires.

Une navigation homogène

L'accessibilité d'un site se mesure aussi par l'homogénéité et la cohérence de construction et de structuration des pages. Les efforts liés à la structuration auront un impact direct sur la facilité de navigation et de lecture à travers les pages du site.

◄ Figure 6.14 : *Le menu*

La barre du menus en version texte peut être répétée en haut ou en bas. Le développeur peut imaginer une lecture séquentielle de son site comme une visite guidée avec des liens vers la page principale puis celle qui suit dans un ordre donné pour donner le plus de sens possible à cette navigation. Ainsi la page principale peut ensuite conduire à la page de nouvelles, puis à la page du dernier article, etc. Les liens peuvent être séparés par un caractère sans signification comme | (sur le clavier du PC, il se trouve au-dessous du 6).

Quand une page est longue, l'insertion de liens internes facilite l'accès à des parties précises de la page.

Si une page est trop lourde, il vaut mieux la diviser en plusieurs pages par égard pour tous ceux qui se connectent encore par modem. Le poids de la page est composé du poids du code source et de tous les fichiers périphériques afférents, comme les images, les fichiers JavaScript, les fichiers son, les feuilles de style, etc. Il est conseillé d'éviter les pages qui font plus de 70 Ko et d'indiquer le poids de l'image que l'utilisateur s'apprête à afficher en cliquant sur la miniature correspondante.

Les touches d'accès

Les touches d'accès pour accéder aux liens sont très pratiques. Tous les graphistes ou maquettistes professionnels le savent : utiliser les touches de raccourci va beaucoup plus vite que d'utiliser la souris. Les `accesskey` fournissent ces touches de raccourci dans un site pour naviguer d'une page à l'autre. La plupart des navigateurs y obéissent. Les utilisateurs qui n'utilisent pas un navigateur audio ne s'en aperçoivent pas. Une solution est de faire commencer l'intitulé de chaque lien avec une lettre différente et de s'en servir comme touche (toujours combinée avec Alt ou Alt+Maj ou Ctrl selon le navigateur).

```
<a href="accueil.htm" accesskey="a"><em>A</em>ccueil</a>
<a href="catalogue.htm" accesskey="c"><em>C</em>atalogue</a>
<a href="magasin.htm" accesskey="m"><em>M</em>agasin</a>
<a href="nous_contacter.htm" accesskey="n">
<em>N</em>ous contacter</a>
<a href="forum.htm" accesskey="f">Notre <em>F</em>orum</a>
```

Il est aussi possible de mettre des chiffres (ceux en haut du clavier, non ceux du clavier numérique) comme touches d'accès.

```
<a href="accueil.htm" accesskey="1">Accueil</a>
<a href="catalogue.htm" accesskey="2">Catalogue</a>
<a href="magasin.htm" accesskey="3">Magasin</a>
<a href="nous_contacter.htm" accesskey="4">Nous contacter</a>
<a href="forum.htm" accesskey="5">Forum</a>
```

Une astuce intéressante est d'afficher automatiquement les `accesskey` après le lien pour en faciliter l'usage. Ici les touches d'accès sont inscrites entre crochets après le lien. Cela ne fonctionne pas pour Internet Explorer.

```
a:after {
content: " [" attr(accesskey) "] ";
   }
```

Le site `http://openweb.eu.org/articles/accesskey_essai_non_transforme` explique les procédures dans un article mettant en avant que les `accesskey` sont une méthode non aboutie.

Pour le site Écoute S'il Pleut, nous l'avons utilisée.

```
div.liens {
 background-color: #FFFFE1;
 padding: 10px;
 color: #312E67;
 text-align: center;
 clear: both;
 font-size: 0.75em;
}
div.liens a{
color: navy;
}
div.liens a:hover{
text-decoration: none;
background-color: navy;
color: yellow;
}
.liens a:after {
content: "[" attr(accesskey) "]";
   }
```

Ici la technique mise en œuvre consiste à générer le contenu avec le pseudo-élément `:after`. La lettre qui sert de touche d'accès apparaît après le lien (voir Figure 6.15).

```
<div class="liens">
Accueil
<a href="gites.php" accesskey="B" title="Les Gîtes"></a>
<a href="#" accesskey="C" title="Venir"></a>
<a href="#" accesskey="D" title="Visites"></a>
<a href="#" accesskey="E" title="Notre Projet"></a>
<a href="#" accesskey="F" title="Liens amis"></a>
<a href="#" accesskey="G" title="Contact"></a>
<a href="#" accesskey="H" title="Les pépinières"></a>
<a href="#" accesskey="I" title="Le Conseil"></a>
<a href="#" accesskey="J" title="Nos Partenaires"></a>
</div>
```

◄ Figure 6.15 :
Le menu avec les touches d'accès sous Firefox

Le lien vers la page courante est désactivé (ici c'est la page **Accueil**) tandis que les balises de lien n'ont pas de contenu. Ainsi, sur les navigateurs non conformes comme Internet Explorer, seul le titre de la page apparaît, tandis que sur les autres, ce titre est accompagné des touches d'accès entre crochets et soulignées. Ces touches d'accès sont explicites et donnent plus de facilité à ceux qui veulent les utiliser.

◄ Figure 6.16 :
Le menu avec les touches d'accès sous Internet Explorer

6.12 L'ordre de lecture

L'affichage des éléments se fait dans l'ordre où ils sont cités dans le code source. De la même façon, avec un navigateur audio, les éléments sont lus au fur et à mesure qu'il décode le code source. L'avantage avec le code CSS est qu'il est théoriquement possible de placer les blocs à l'endroit qui convient sans se soucier de l'endroit où ils sont évoqués dans le code source. Ainsi, l'écriture du code source peut d'abord prendre en compte une hiérarchie d'informations et commencer par la plus importante.

Qu'est-ce qui est le plus important après tout le contenu d'en-tête ? Le visiteur vient chercher le contenu sur une page, le lit, puis cherche des liens pour passer à une autre page. C'est pourquoi, il est préférable de placer le conteneur `div`, qui contient le texte de la page, avant l'autre conteneur `div`, qui renferme le contenu de la palette de navigation.

En attendant que tous les navigateurs soient conformes aux recommandations, le développeur peut insérer un lien invisible qui amène le visiteur directement au contenu, en court-circuitant les liens de navigation.

```
<link rel="contenu" title="contenu de la page" href="#contenu" />
```

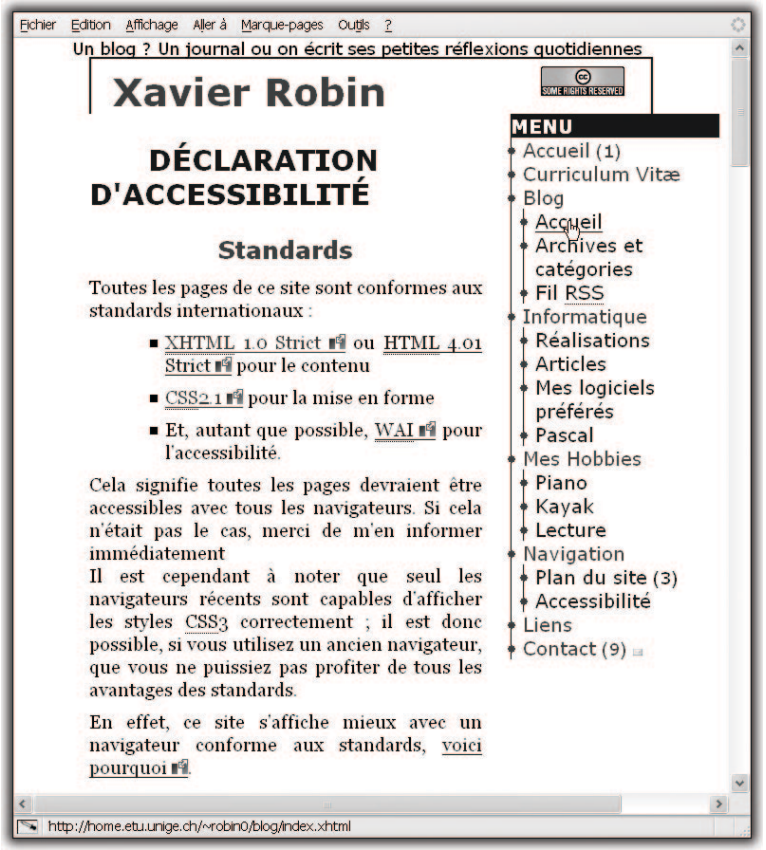
6.13 Les aides

Les aides sur un site démontrent la volonté du développeur de prêter attention aux internautes.

La page d'aide

Elle peut expliquer les touches d'accès et les combinaisons de touches selon les navigateurs, contenir une FAQ pour aider à l'interactivité, afficher la signification des icônes présentes sur les pages. Elle évite de donner des explications fastidieuses dans les pages courantes, de les répéter et de se creuser la tête pour savoir où les placer. Sur cette page, le développeur a l'opportunité de demander le concours du lecteur pour relever les erreurs et faire des suggestions. Elle est un mode d'emploi pour expliquer au visiteur la meilleure façon d'exploiter le site.

La page de Xavier Robin est un exemple de bonnes pratiques pour la page d'aide (`http://home.etu.unige.ch/~robin0/accessibilite.html`).



▲ Figure 6.17 : *Page d'aide du blog de Xavier Robin*

Les aides invisibles de navigation

La balise `link` permet d'insérer dans la page des liens invisibles de navigation qui sont, cependant, perçus par les navigateurs audio et permettent au besoin de court-circuiter la page. Le développeur peut faire une sorte de circuit de son site pour les personnes handicapées.

```
<link rel="home" title="Accueil" href="http://url/de/la/page/accueil" />
<link rel="prev" title="Titre de la page précédente"
```

```
      href="http://url/de/la/page/precedente" />
<link rel="next" title="Titre de la page suivante"
      href="http://url/de/la/page/suivante" />
```

6.14 La page nue

La page HTML exploite un certain nombre de technologies d'affichage, comme le JavaScript, les CSS. Avez-vous essayé votre site en désactivant toutes ces technologies ? Est-il encore possible d'afficher un contenu lisible ? Est-il encore possible de naviguer, de valider les formulaires et de profiter des différentes fonctionnalités du site ?

Pour ce qui relève de JavaScript (validation de formulaires, ouverture des fenêtres…), il est possible de proposer une alternative.

Le bouton de soumission d'un formulaire ou l'image qui sert de bouton de soumission ont une signification alors qu'une balise `button` avec un événement `onClick` est dénuée de tout contexte qui permet de comprendre sa fonctionnalité.

Une fenêtre pop-up peut être ouverte avec un lien doté d'une propriété `target="blank"`.

L'utilisation d'un navigateur texte met en évidence les incohérences, les images sans légende et les liens non explicites.

6.15 Conclusion

Un jour, l'accessibilité sera mise en œuvre systématiquement et tout naturellement. Toutes les recommandations seront entrées dans les logiciels de création de site web, qui, par défaut, afficheront les balises d'accessibilité à intégrer.

Tenir compte de l'accessibilité permet d'appliquer nombre de bonnes pratiques, de bonnes habitudes qui changent la manière de concevoir un site et aussi oblige à travailler plus professionnellement.

Intégrer l'accessibilité est une façon de montrer la maturité du développeur mais aussi du Web.

7

Le ravalement de façade

Il peut être judicieux de refaire un site qui a été initialement créé sans feuille de style. S'il a perdu de son attractivité, avant de perdre ses visiteurs fidèles et de rebuter les nouveaux, il est urgent d'opérer un ravalement avant qu'il ne devienne obsolète.

7.1 Quand opérer une refonte d'un site ?

Il existe plusieurs outils pour savoir si le moment est opportun de modifier l'apparence d'un site. En général, la refonte visuelle remet en question le mode de communication du site et son contenu. Le style du contenu sera aussi modifié. Un style visuel tonique appelle des textes plus courts, des termes plus directs, une structure plus percutante, plus d'images. La décision de refonte s'appuie obligatoirement sur une stratégie nouvelle, une approche différente de communication avec les internautes.

Le site Jolies Pages expose des refontes de sites sur `www.joliespages.com/pages/00changegraphe.php`.

▲ Figure 7.1 : *Jolies pages*

La baisse de fréquentation du site web et la désaffection des internautes fidèles sont des signaux révélant qu'il est temps de faire le point et de monter un nouveau projet. Faire appel à un graphiste sans avoir travaillé sur le concept du site est peine perdue car le résultat ne sera qu'un maquillage ou un cache-misère qui ne ramènera pas les visiteurs.

Il est peut-être intéressant de procéder à un sondage et de lancer un appel aux visiteurs pour qu'ils participent au projet. Ensuite, il s'agira de communiquer dans la transparence au fur et à mesure de l'avancement du projet. Il faut un vrai renouveau et non pas un simple changement de look.

Si les échos des visiteurs du site révèlent que le site s'affiche mal avec certains navigateurs, il faudra rafraîchir l'interface pour qu'elle soit compatible. C'est parfois l'opportunité pour faire une vraie refonte du site et utiliser éventuellement les feuilles de style.

Quand un site est conçu, il est composé d'un certain nombre de pages et de liens de navigation. Ensuite les besoins évoluent, des pages s'ajoutent. Parfois le centre d'intérêt du site se déplace sur un thème qui était traité en périphérie. La refonte du site est alors nécessaire pour mettre le nouveau thème en avant et clarifier la communication.

Un site initié par une personne grossit, demande de plus en plus de temps pour la mise à jour. Le développeur des débuts est de plus en plus dépassé par son projet, qui demande alors à se professionnaliser. La gestion du site doit être confiée à une équipe et les mises à jour rationalisées. C'est le moment d'utiliser les CSS et le XHTML, pour qu'il soit possible de modifier les pages sans y passer trop de temps.

Un site peut être mal référencé parce qu'il s'affiche mal ou parce que les informations importantes ne sont pas mises en évidence, ou encore parce qu'il compte moins de cinq pages. Un site conforme aux directives du W3C sera mieux référencé car le robot collecteur à la solde des moteur de recherche saura où aller chercher l'information à indexer.

7.2 Où commencer ?

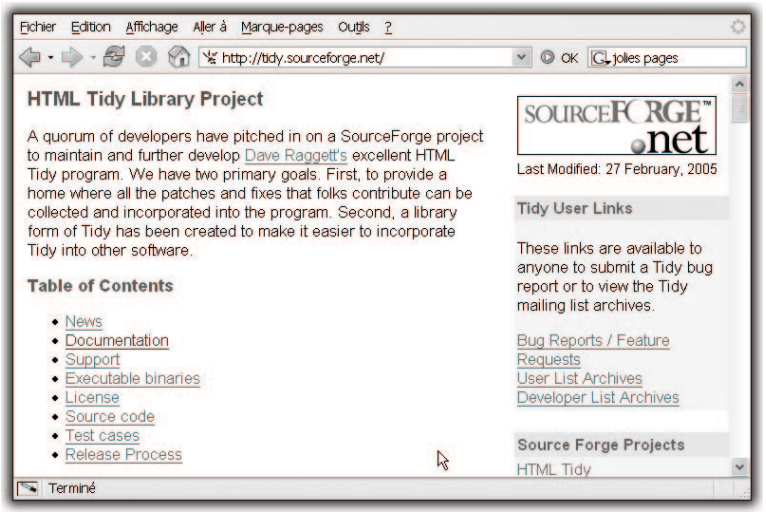
Deux possibilités s'offrent au candidat à une refonte de site :

- Soit il commence par normaliser le site en transformant le HTML en XHTML et en esquissant une feuille de style.

- Soit il recommence le site et récupère le contenu une fois la charte graphique refaite.

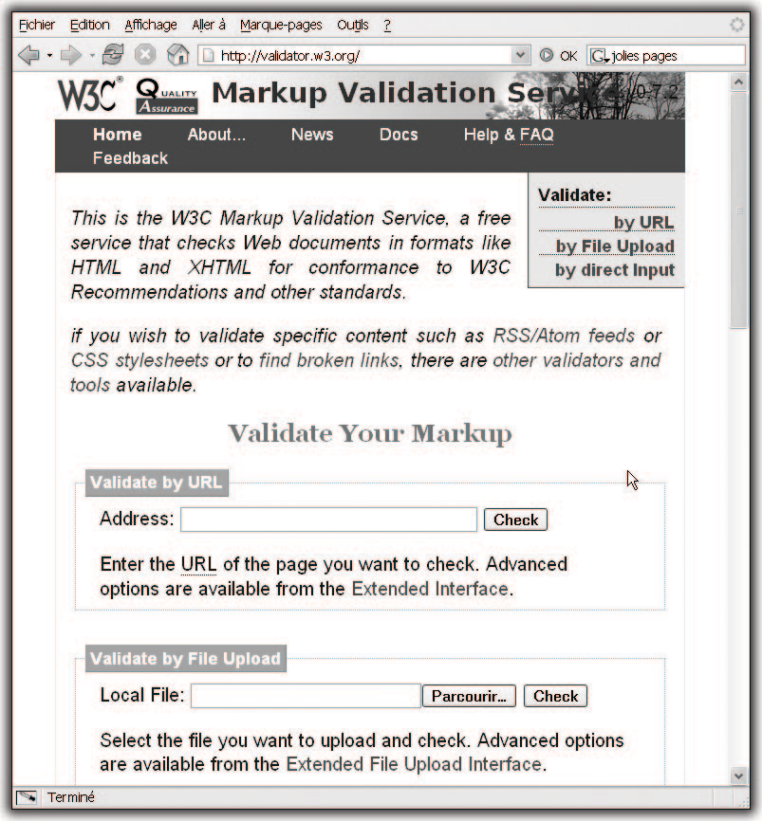
Pour transformer du HTML en XHTML, il existe des outils comme Tidy. Tidy est un nettoyeur de code, mais il ne traite que du code qui n'est pas trop sale. Au fur et à mesure des versions, il devient de plus en plus complet et peut par exemple vérifier si la page répond aux impératifs de l'accessibilité ou nettoyer un fichier web généré par Word 2000.

Le nettoyage des balises comprend l'ajout de la barre oblique dans la balise fermante ou la correction de la balise fermante si elle est mal intitulée. Les chevauchements de balises sont rectifiés. Des guillemets sont ajoutés pour encadrer les valeurs des propriétés. Ce logiciel miracle (`http://tidy.sourceforge.net`) a bien d'autres fonctionnalités et il est souvent implémenté dans d'autres logiciels, comme HTML Tidy (`www.framasoft.net/article1010.html`).



▲ Figure 7.2 : *Tidy*

Pour finir la transformation, entrez l'URL de vos pages dans le validateur du W3C sur `http://validator.w3.org`.



▲ Figure 7.3 : *Validator*

Une fois que le fichier source a été transformé en XHTML propre, l'auteur peut transposer certaines propriétés dans des sélecteurs de la feuille de style pour simplifier encore le source. Ensuite une question se pose : jusqu'à quel point faut-il transformer le fichier source ?

7.3 La question des tableaux

Tous les sites créés sans Flash ou sans feuille de style utilisent des tableaux pour structurer leurs pages. Faut-il conserver ces tableaux ou les

transformer en conteneurs ? Il faut essayer la seconde option car les tableaux sont compliqués à maintenir dès qu'une mise à jour s'impose, surtout s'ils sont imbriqués. Les feuilles de style, si elles sont bien documentées, sont faciles à comprendre car le développeur a une vue générale du code de la page, sans que viennent interférer les détails. Il comprendra tout de suite la philosophie du design et passera moins de temps à comprendre la raison de la présence de tel ou tel code.

7.4 La refonte du site photo-de-classe.com

Le site photo-de-classe.com a été créé en 1999 sur une idée de Christophe Bovio. Il l'a conçu, réalisé et maintenu depuis la création. Ce site, pionnier dans son domaine, a été repris maintes fois depuis. Le concept est le suivant : permettre aux internautes de retrouver leurs camarades d'école. Pour cela, il suffit de s'inscrire et de répertorier son parcours scolaire, grâce à un formulaire et une liste d'écoles, de collèges, de lycées, d'universités et autres grandes écoles, mise à jour régulièrement. Le site n'a pas changé depuis ce temps, notamment au niveau de l'interface et des fonctionnalités présentées à l'internaute. Christophe Bovio a simplement ajouté le concept des "bouteilles à la mer". Il s'agit d'un système qui permet à un membre de lancer un appel à tous les autres membres. Ces appels sont envoyés automatiquement selon une périodicité donnée (voir Figure 7.4).

Le site est hébergé chez OVH et développé en PHP3 puis avec la version 4 et avec une base de données MySQL 3.5, qui n'a pas changé depuis les débuts et qui fonctionne très bien.

Les modifications ont porté sur la maintenance car le serveur subit une charge de plus en plus forte au fur et à mesure de l'affluence des internautes, suite à des passages à la télévision et des articles dans des magazines à grand tirage. Tout le travail au cours des années passées a été d'améliorer le back-office, c'est-à-dire les interfaces, pour administrer le site et déclencher les événements. Pour réduire la charge du serveur, il a fallu aussi gérer le cache, modifier les applications gourmandes en microprocesseur et alléger le code. Christophe s'est rendu compte que la

moitié du code seulement était vraiment utile et que de nombreuses lignes était dévolues à des tables imbriquées qui rendaient le source illisible (voir Figure 7.5).

▲ Figure 7.4 : *Photo-de-classe.com avant*

▲ Figure 7.5 : *Photo-de-classe.com après*

Christophe Bovio a fait appel à une graphiste qui a complètement rafraîchi l'aspect du site en gardant une charte proche de l'original. Le site puise dans l'univers BD pour parler à son public.

▲ Figure 7.6 : *Univers BD*

Le choix des CSS s'est imposé pour plusieurs raisons :

- alléger le code pour le maintenir plus facilement dès qu'il s'agit de faire une modification ;
- réduire la bande passante consommée, avec un code allégé ;
- adapter plus aisément la présentation aux principales célébrations de l'année ;
- rendre le site accessible à tous ;
- être mieux référencé dans les moteurs de recherche car si le développeur recherche le code utile, il en est de même des moteurs de recherche, qui écument les sites afin d'en extraire l'information utile.

La mise en CSS a confronté Christophe à quelques soucis. Le chemin entre le graphisme et la mise en (X)HTML et en CSS est semé d'obstacles, surtout s'il faut disposer de nombreuses boîtes dans la page et que tout s'affiche quelle que soit la définition d'écran.

Un petit graphisme sympathique se dévoile progressivement sur les écrans ayant une résolution au-delà de 800 × 600 pixels. Le site est centré sur un fond uni bleu mauve (`#669`) et construit sur des mesures précises.

▲ Figure 7.7 : *Dessins cachés*

Parfois, la construction d'un site ressemble à l'édification d'un château de cartes : vous posez les cartes de plus en plus haut, vous ajoutez la dernière carte et la fragile construction s'écroule. La difficulté augmente potentiel-

lement selon le nombre de boîtes imbriquées, puis la fragilité du tout augmente avec l'interprétation de chaque navigateur.

Dans son site, Christophe propose une navigation par onglets. Ces onglets sont faits selon la méthode des portes coulissantes créée par Douglas Bowman (`http://pompage.net/pompe/portescoulissantes/`).

▲ Figure 7.8 : *Pompage.net*

Le nouveau site est beau et sympathique, avec une pointe d'humour qui crée un climat de complicité entre tous les membres inscrits (gratuitement) ou même les simples visiteurs. Le fond blanc et la découpe claire des différentes rubriques en page d'accueil créent un espace bien structuré, où

la navigation est aisée. Les arrondis des boîtes, les sourires des enfants farceurs et le sourire franc et plein d'humour de l'animateur-webmaster-développeur sont engageants. Un nouvel arrivant sera tenté de s'inscrire en toute transparence.

◄ Figure 7.9 :
Petit personnage

Pour que le site garde sa marque de fabrique, la couleur mauve dominante est reconduite et la photo du webmaster est là pour assurer la continuité. L'esprit de ce site se poursuit et la photo du webmaster atteste de la transparence et de l'éthique que photo-de-classe.com a su garder et auxquelles s'est engagé Christophe Bovio.

L'évolution du site

La feuille de style de photo-de-classe.com a beaucoup évolué entre les deux moutures du site. La première feuille de style a permis d'enlever les indications de forme à l'intérieur du HTML et de les reporter dans la feuille de style. Tout est inséré dans des tables sans que des boîtes soient véritablement positionnées.

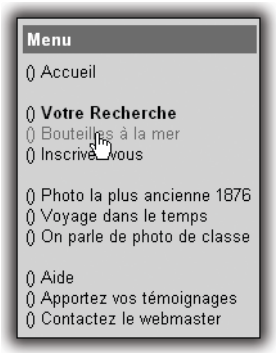
L'une des premières actions à effectuer est de reporter dans une feuille de style les indications de forme insérées dans le code HTML. Voici une partie du code :

```
input {
   font: 8pt Arial, helvetica, sans-serif
}
select {
   font: 8pt Arial, helvetica, sans-serif
}
textarea {
   font: 8pt Arial, helvetica, sans-serif
}
button {
   font: 8pt Arial, helvetica, sans-serif
}
a {
```

```
   color: #000000
}
a:hover {
   color: #666699
}
a.black {
   color: #000000
}
a.black:hover {
   color: #CCCCFF
}
a.blackBold {
   font: bold 8pt Arial,helvetica,sans-serif;
}
a.blackBold:hover {
   font: bold 8pt Arial,helvetica,sans-serif;
}
a.white {
   color: #FFFFFF; text-decoration: none;
}
a.white:hover {
   color: #CCCCFF; text-decoration: none;
}
a.promo {
   font: 8pt Arial, helvetica, sans-serif; color: #000000;
text-decoration: none;
}
a.promo:hover {
   color: #666699;
}

a.y {
   font: 8pt Arial, helvetica, sans-serif; color: #000000;
text-decoration: none;
}
a.y:hover {
   color: #666699;
}

a.invisible {
   text-decoration: none;
}
a.invisible:hover {
   text-decoration: none;
}

a.bouton {
   font: 9pt Arial,helvetica,sans-serif;
   color: #000000;
```

```
   background-color: #CDCBFC;
   text-decoration: none;
   text-align: center;
   padding: 5px;
   border: 2px outset #ffffff;
}

a.bouton:hover {
   color: gray;
   background-color: #CDCBFC;
   border: 2px inset #ffffff;
}

a.nbinscrits {
   font: bold 9pt Arial,helvetica,sans-serif; color: #FFFFFF;
}

input.form {
background-color: #ccccff; font: 8pt Arial, helvetica,
sans-serif;
}
textarea.form {
   background-color: #ffffff; font: 8pt Arial, helvetica,
sans-serif;
}
select.form {
   border-right: #c2c3c2 1px solid;
   border-top: #c2c3c2 1px solid;
   border-left: #c2c3c2 1px solid;
   border-bottom: #c2c3c2 1px solid;
   background-color: #ccccff
}
checkbox.form {
   border-right: #c2c3c2 1px solid;
   border-top: #c2c3c2 1px solid;
   border-left: #c2c3c2 1px solid;
   border-bottom: #c2c3c2 1px solid;
   background-color: #ccccff
}
td.invert_header {
   font: bold 8pt Arial,helvetica,sans-serif;
   color: #FFFFFF;
   background-color: #666699;
}
td.GrosHeader {
   font: bold 12pt Arial,helvetica,sans-serif;
   color: #000000;
   background-color: #CCCCFF;
   text-align: center
```

```
}
td.NavigationPages {
   font: 8pt Arial,helvetica,sans-serif;
   color: #000000;
   background-color: #CCCCFF;
   text-align: center
}
td.text {
   font: 8pt Arial,helvetica,sans-serif;
   color: #000000;
   background-color: #ccccff
}
```



◄ Figure 7.10 : *Liens*

Dans le nouveau site, les compteurs sont tout de suite remis à 0 pour effacer les réglages par défaut des navigateurs.

```
html, body {
   padding: 0;
   margin: 0;
   background-color: #666699;
   font: 9px verdana, arial,sans-serif;
   height: 100%
}

* {margin:0; padding:0;}
```

Le parti est pris de superposer les fonds pour un dévoilement progressif des différents motifs qui les constituent.

```
/*************************************************************
Système de fonds superposés masquables
*************************************************************/
#page {
background: url(page_bg_motif.gif) repeat-y center center;}
#page_pifpaf {
background: url(page_bg_pifpaf.gif) no-repeat center top;}
#page_780 {
background-color: #FFFFFF;
width: 780px;
margin: 0px auto;
}
```

▲ Figure 7.11 : *Image de fond bandeau_p2c_annees70.gif*

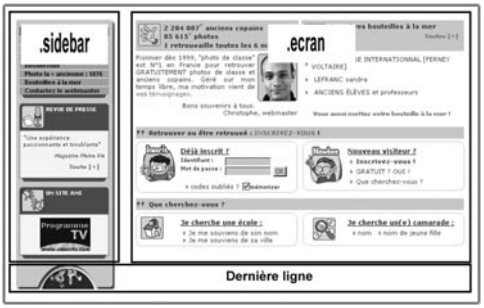
```
/*************************************************************
Header avec pub 468 × 60
*************************************************************/
#page_780_header {
background: url(/images/bandeau_p2c_annees70.gif) #FFFFFF
no-repeat center top; width: 780px; height: 245px;}
#bandeau_468_60 {
position: absolute;
margin-top: 27px;
margin-left: 301px;
float: right;
width: 468px;
height: 60px;
}
#page_780_header_light {
background: url(/images/bandeau_p2c_light.gif) #FFFFFF
no-repeat center top;
width: 780px;
height: 113px;
}
```

L'en-tête est placé en position absolue et tout est tracé au cordeau. Le pied de page est déclaré juste après.

```
/*************************************************************
Pied de page simple
*************************************************************/
#page_780_footer  {
clear: both;
background: url(/images/page_footer.gif)
no-repeat center bottom;
width: 780px;
height: 45px;
}
#page_780_footer img  {
display: inline;
vertical-align: middle;
}
#page_780_footer p  {
position: absolute;
margin-top: 4px;
margin-left: 10px;
margin-right: 10px;
font-size: 9px;
color: #FFFFFF;
text-align: left;
}
#page_780_footer p a {
color: #FFFFFF;
text-decoration: none;
}
```

▲ Figure 7.12 : *Image de fond page_footer.gif*

Sous l'en-tête, se déroule le menu à onglets, puis se déploie une table à deux colonnes et deux lignes.



◄ Figure 7.13 : *Table avec la classe de chaque colonne et ligne*

Pour le menu horizontal, la structure choisie, comme dans la plupart des cas, est celle des listes non ordonnées sans puce.

```
<LI><a href="/actus.php" onMouseover="expandcontent('sc6')">
<span>Medias</span></a>
</LI>
```

La partie du code source s'y référant pour les sous-menus est la suivante :

```
<div id="sc6" class="submenubarcontentR">
<a href="/recommander.php"
title="Merci de contribuer à faire connaître mon site">
Faire connaître ce site</a>
   <a href="/actus.php?media=tv">Télévision</a>
   <a href="/actus.php?media=radio">Radio</a>
   <a href="/actus.php?media=magazine">Magazines</a>
   <a href="/actus.php?media=quotidien">Quotidiens</a>
   <a href="/actus.php?media=web">Internet</a>
   <!--<a href="#">Accès journalistes</a> -->
       </div>
```

Le code dans la feuille de style concernant le menu est plus complexe.

```
/*************************************************************
Barre de menus horizontale
*************************************************************/
#menuhor {
   position: relative;
   height: 33px;
   width: 770px;
   background-color: transparent;
   margin-top: -10px;
}

#menuhor ul {
   padding-right: 10px;
   padding-left: 10px;
   padding-bottom: 0px;
   margin: 0px;
   padding-top: 10px;
   list-style-type: none;
}

#menuhor li {
   display: inline;
   padding-right: 0;
   margin: 0;
```

```
}

#menuhor a {
   padding-right: 0px;
   padding-left: 9px;
   background: url(/images/left_both.gif)
no-repeat left bottom;
   background-position: 0px -127px;
   float: left;
   padding-bottom: 0px;
   margin: 0px;
   padding-top: 0px;
   text-decoration: none;
   margin-right: 3.5px;
}
#menuhor a span {
   padding-right: 15px;
   display: block;
   padding-left: 6px;
   background: url(/images/right_both.gif)
no-repeat right bottom;
   background-position: 100% -126px;
   float: left;
   padding-bottom: 4px;
   font-size: 12px;
   font-weight: bold;
   color: #FFFFFF;
   padding-top: 5px;
}

#menuhor a span {
   float: none;
}

#menuhor a:hover span {
   color: #FFFFFF;
}

#menuhor #current a {
   background-position: 0px -275px;
}

#menuhor #current a span {
   background-position: 100% -275px;
   color: #FFFFFF;
}

#menuhor a:hover {
   background-position: 0% -275px
```

```
}

#menuhor a:hover span {
   background-position: 100% -275px
}

/*************************************************************
Barre de menus horizontale : catch entrée/sortie dans la barre
*************************************************************/
#mouseouttop
{
   position: relative;
   height: 5px;
   z-index: 100; /* pour un passage assuré */
}

#mouseoutbottom
{
   position: relative;
   height: 5px;
   z-index: 100; /* pour un passage assuré */
}

/*************************************************************
Barre de menus horizontale : sous-menus
*************************************************************/
#submenubar
{
   background-color: #FFFFFF;
   position: relative;
   height: 30px;
}

.submenubarcontentR{
   display:none;
   margin-right: 24px;
   margin-left: 4px;
   margin-top: 6px;
   text-align: right;
}

.submenubarcontentL{
   display:none;
   margin-right: 4px;
   margin-left: 11px;
   margin-top: 6px;
   text-align: left;
}
```

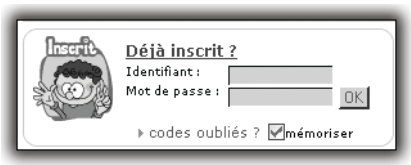
```
.submenubarcontentbandeau {
   display:none;
   margin-right: 24px;
   margin-left: 16px;
   margin-top: 6px;
   text-align: left;
}

.submenubarcontentR a:link, .submenubarcontentR a:visited,
.submenubarcontentL a:link, .submenubarcontentL a:visited {
   text-decoration: none;
   padding-left: 4px;
   padding-right: 4px;
   border-right: 1px solid #000000;
   font-size: 10px;
   font-weight: bold;
   color: #110B05;
}

.submenubarcontentR a:hover, .submenubarcontentL a:hover{
   color: #FF6600;
}

/*************************************************************
Zone de messages : actualités, infos, astuces, etc.
*************************************************************/
.bandeautxt {
   font-size: 10px;
   font-weight: bold;
   color: #655755;
   margin-right: 5px;
}

.bandeautxt span {
   color: #404069;
   padding-left: 5px;
   padding-right: 5px;
   border-bottom: 1px solid black;
}

.bandeautxt a:link, .bandeautxt a:visited {
   font-size: 10px;
   font-weight: bold;
   color: #FF6600;
   text-decoration: none;
   margin: 2px 10px 2px 2px;
   border: none;
}
```

```
.bandeautxt a:hover {
   color: #655755;
}
```

Plusieurs fichiers JavaScript sont ajoutés pour permettre le déploiement de tous les menus au passage de la souris.

```
<link rel="stylesheet" type="text/css"
href="/images/styles_bandeau_p2c.css"
media="screen, projection, print, tv" />

<script src="js/bandeautxt.js" type="text/javascript"
  language="javascript"></script>
<script src="js/submenu.js" type="text/javascript"
  language="javascript"></script>
<script src="js/bouteilles.js" type="text/javascript"
  language="javascript"></script>
<script src="js/468rotate.js" type="text/javascript"
  language="javascript"></script>
<script src="js/utils.js" type="text/javascript"
  language="javascript"></script>

<link rel="shortcut icon" type="image/bmp"
href="http://www.photo-de-classe.com:8888/favicon.ico" />
```

Voici le code pour les éléments de formulaire.



◄ Figure 7.14 : *Formulaire d'inscription*

```
/*************************************************************
Éléments de formulaire
*************************************************************/

form.inscrits {
   margin: 0;
   padding: 0;
   width: 195px;
   line-height: 100%;
}
.inscrits label {
   color: #110B05;
   width: 80px;
```

```
   float: left;
}
.inscrits input.type_text, .recherche input.type_text {
   background-color: #CCCCFF;
   border-top: 1px #110B05 solid;
   border-left: 1px #110B05 solid;
   border-right: 1px #FFFFFF solid;
   border-bottom: 1px #FFFFFF solid;
   font-size: 9px;
   color: #110B05;
   width: 80px;
   /*height: 16px;*/
}
.bouton {
   border: 2px solid #343441;
   border-top-color: #FFFFFF;
   border-left-color: #FFFFFF;
   background-color: #CCCCFF;
   color: #655755;
   font-size: 10px;
}
.bouton:hover {
   color: #6363A0;
   border: 1px inset #74747F;
}
```

Voici une partie du code source :

```
<div id=blockL>
<DIV class=boite><DIV class="boite_br"><DIV class="boite_tl">
<DIV class="boite_tr">
<div class="picto_gd"><img src="images/picto_inscrit.gif"
width="65" height="67" alt="" /></div>
<div class="colonne">
<div class="titreBleu">Déjà inscrit ?</div>
<form class="inscrits" name="identification_form"
action="/mesclasses.php" method="post">
<input type="hidden" name="identification" value="oui" />
<label for="identifiant">Identifiant : </label>
<input class="type_text" id="identifiant"
name="identification_login" type="text" size="10"
maxlength="80" />
<br />
<label for="password">Mot de passe : </label>
<input class="type_text" id="password"
name="identification_password" type="password"
size="10" maxlength="80" /> 
<input class="bouton" type="submit" name="OK" value="OK" />
<br />
```

```
<div class="spacer"> </div>
<ul class="inline"><li class="inline">
    <a class="soft" href="/perte_mail.php">
    codes oubli&eacute;s ?</a>
</li></ul> 
<input type="checkbox" name="memocodes" value="1" checked />
mémoriser
</form>
</div>
     </DIV></DIV></DIV></DIV>
</div>
```

Voici maintenant le code CSS qui permet d'avoir les boîtes à coins arrondis.

```
/*************************************************************
Boîtes à coins arrondis
*************************************************************/

.boite {
background: url(/images/round_bl.png) no-repeat left bottom;
width: 289px;
}
.boite_br {
background: url(/images/round_br.png) no-repeat right bottom
}
.boite_tl {
background: url(/images/round_tl.png) no-repeat left top
}
.boite_tr {
padding: 5px;
background: url(/images/round_tr.png) no-repeat right top;
}

.colonne {
margin-left: 75px;
}
```

Plusieurs boîtes sont superposées pour accueillir chacune un coin arrondi. Le contenu des différentes rubriques est disposé dans des boîtes flottantes.

```
/*************************************************************
Placement des divers pictos et photos
*************************************************************/
.picto {
 float: left;
 display: block;
```

```
 width: 40px;
 text-align: center;
 margin: 4px;
 margin-top: 2px;
}
```

Les tableaux de données ne sont pas bannis, mais ils sont faits proprement afin de faciliter la tâche des navigateurs audio. Il faut éviter d'être orthodoxe et de tomber dans l'excès qui consiste à n'utiliser que des boîtes `div` positionnées en CSS. Il est préférable de s'adapter pour avoir un beau site sans problème.

Quelles que soient les options, elles demandent une grande cohérence. Ici c'est le cas. Les boîtes `header` et `footer` sont en position absolue, la boîte principale en position relative, puis les rubriques en position flottante. Tout se tient : de l'option de départ découlent les autres options, même si chaque créateur garde une certaine liberté au fur et à mesure de l'édification du site et de son architecture.

La refonte d'un site peut être longue. Il est prudent de ne pas annoncer de date dans un répertoire du site avant que le travail ne soit achevé. Mais une fois la mise en CSS réalisée, les refontes suivantes sont plus simples à mettre en place car souvent, elles ne concernent que des éléments graphiques remis au goût du jour et insérés dans l'architecture originelle.

8

Vers la CSS3

Le futur des feuilles de style est la norme CSS3. La feuille de style a désormais des possibilités qui la rapprocheront de la PAO et donc du papier.

Tous les bricolages et bidouillages nécessaires jusqu'à présent pour améliorer le rendu des CSS ne sont plus nécessaires car la CSS3 les incorpore. Plus besoin de se creuser la tête pour avoir un affichage homogène des boîtes sous forme de colonnes, plus besoin de créer des images spéciales pour arrondir les angles !

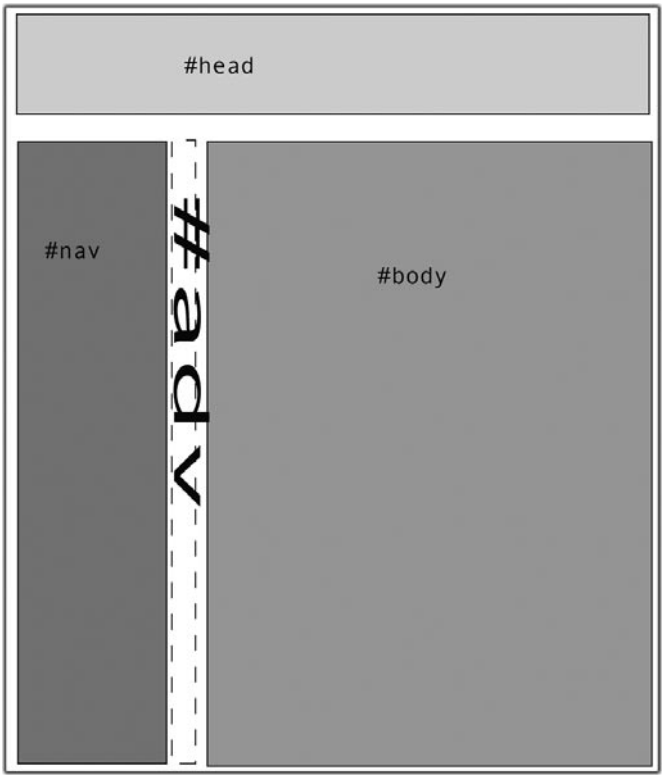
La CSS3 est vraiment la version qui mène la CSS à la maturité. Il reste maintenant aux navigateurs à se conformer à cette norme en intégrant les recommandations du W3C.

Les différentes fonctionnalités implémentées rendent le développement des CSS plus intuitif, ce qui devrait plaire aux graphistes qui n'aiment pas passer trop de temps à développer, et laisser plus de place à la création.

8.1 Placer des boîtes

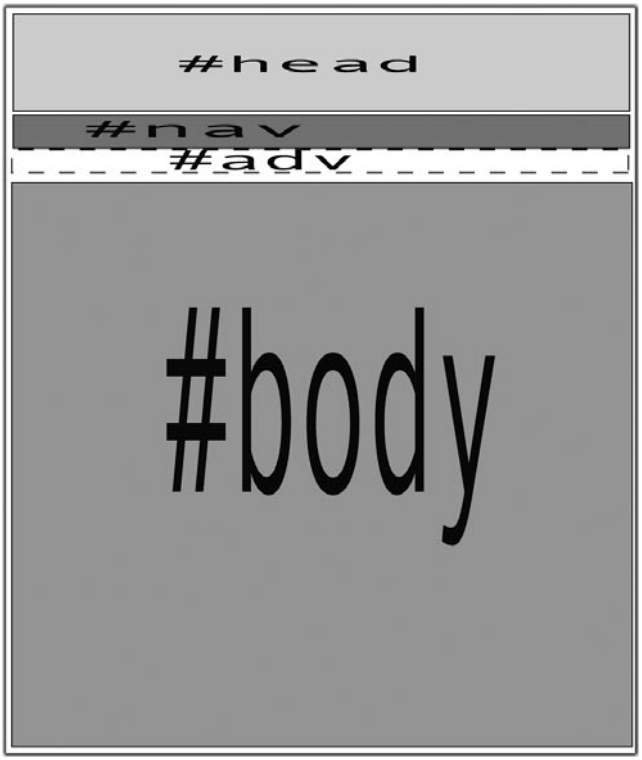
Placer des boîtes devient une tâche naturelle et simple grâce au concept de modèle (*template*). Ainsi une page classique se fait facilement.

```
@media all
{
  body { display: "aaa"
                  "bcd" }
  #head { position: a }
  #nav { position: b }
  #adv { position: c }
  #body { position: d }
}
```



▲ Figure 8.1 : *Résultat*

```
@media all and (max-width: 500px)
{
  body { display: "a"
                  "b"
                  "c" }
  #head { position: a }
  #nav { position: b }
  #adv { display: none }
  #body { position: c }
}
```

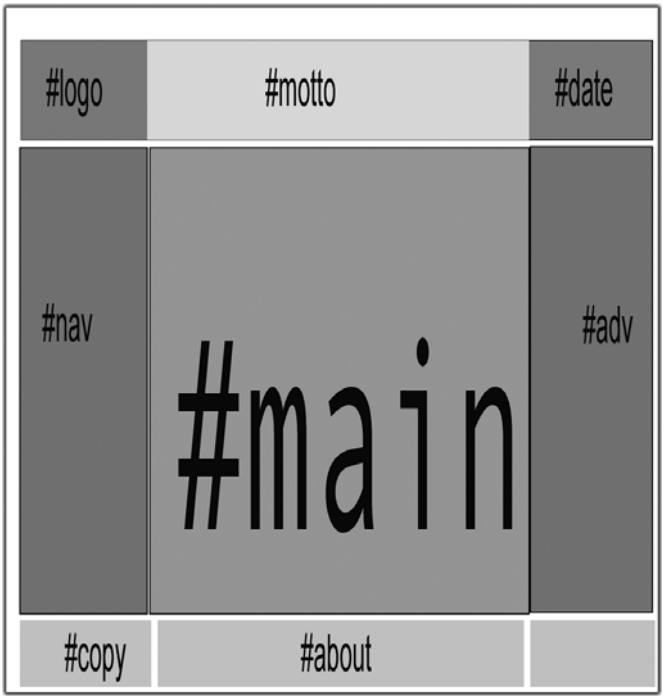


▲ Figure 8.2 : *Résultat*

Ici, avec la propriété `display_model`, il devient possible de créer un modèle qui donne la disposition des différentes boîtes. Chaque boîte est figurée par une lettre. Dans l'exemple précédent, les boîtes sont alignées verticalement.

```
<style type="text/css">
  body {
    height: 100%;
    display: "a.b.c" (2em)
             "....." (1em)
             "d.e.f"
             "....." (1em)
             "g.h.i" (2em)
```

```
            5em 1em * 1em 10em}
  #logo {position: a}
  #motto {position: b}
  #date {position: c}
  #nav {position: d}
  #main {position: e}
  #adv {position: f}
  #copy {position: g}
  #about {position: h}
</style>
<p id=logo><img src=…
<p id=motto>Making Web pages since 1862
<p id=date>August 2, 2004
```



▲ Figure 8.3 : *Résultat*

Les lettres représentent une boîte, les points un espace. De cette façon, la page est presque dessinée par la disposition de ces caractères. Dans cet exemple, la page est composée de trois rangs et de trois colonnes, séparés

par un espace blanc de 1 em. Le rang et la colonne du milieu ne sont pas définis ou définis par un astérisque, ce qui les rend flexibles au niveau de leurs proportions. Les mesures, à la fin de la propriété `display`, donnent la largeur de chaque colonne, la première de 5 em et la dernière de 10 em.

Il est facile aussi d'utiliser des onglets.

```
<style type="text/css">
  body {background: silver; color: black}
  div.records {display: stack; border: outset}
  div.record {display: card}
  h2 {display: tab; width: 5em; border: outset;
      text-align: center}
  h2:current {border-bottom: solid silver}
</style>
<div class=records>
  <div class=record>
    <h2>Men's fashion</h2>
    <ul>
      <li>Oversized jeans, 4 pockets.
      <li>…
    <ul>
  </div>
  <div class=record>
    <h2>Women's fashion</h2>
    …
  </div>
  <div class=record>
    <h2>Children's fashion</h2>
    …
  </div>
</div>
```

Les boîtes ont la possibilité d'avoir des coins arrondis et une ombre, que l'on ne peut obtenir actuellement que par du bidouillage de code.

```
div .nouvelles{
border: thin solid;
box-shadow: 0.2em 0.2em #CCC;
border-radius: 1em;
}
```

`box-shadow` donne les distances par rapport à la boîte, horizontale et verticale.

▲ Figure 8.4 : *Boîte ombrée à bords arrondis*

8.2 Autres caractéristiques

Les couleurs

Il est possible de représenter les couleurs avec le système de codage HSL (*Hue*, *Saturation*, *Luminance*, ou TSL en français pour Teinte, Saturation, Luminance). Ici encore, c'est une manière plus intuitive de définir une couleur, en l'éclaircissant, en lui donnant plus de profondeur, etc.

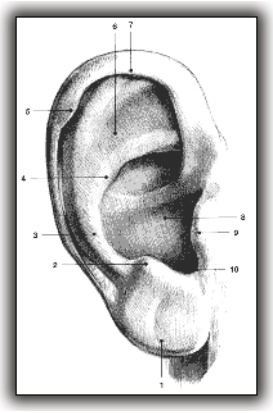
La propriété `transparency` permet de définir une transparence de `0.0` (transparent) à `1.0` (opaque). Les systèmes de codage de couleurs comme RGB ou HSL deviennent RGBA ou HSLA (`A` pour Alpha).

```
p { color: hsla(240, 100%, 50%, 0.5) } /* bleu semi-transparent */
```

Le son

Les caractéristiques du discours pour les navigateurs audio sont définies par diverses propriétés pour le type de voix, le débit du discours, le temps des pauses.

```
h1, h2, h3, h4, h5, h6 {
    voice-family: paul;
    voice-stress: moderate;
    cue-before: url(ping.au)
}
p.heidi { voice-balance: left; voice-family: female }
p.peter { voice-balance: right; voice-family: male }
p.goat  { voice-volume: soft }
```



◄ Figure 8.5 : *Oreille*

Les empreintes de voix sont comme les polices de caractères, elles portent un nom : `paul`, `female`, `male`, etc. La voix `paul` a une accentuation modérée avec un jingle qui l'introduit, comme les annonces de gare. La voix de femme des paragraphes de classe `heidi` vient de la gauche et la

voix d'homme des paragraphes de classe `peter` vient de la droite tandis que les paragraphes de la classe `goat` ont une voix douce.

Les polices

Un certain nombre de caractéristiques étudiées pour la CSS3 concernent les polices et la manière de les afficher. `font-size-adjust` est une caractéristique qui rapproche les CSS de la PAO avec la même précision. Si la police de caractères demandée n'est pas présente sur la machine de l'internaute, une autre police est utilisée ; la différence de taille pour le même corps est compensée : il s'agit de s'ajuster à la taille du caractère prévue.

L'interprétation progressive (*progressive rendering*) est un processus qui consiste en un rendu temporaire des polices en attendant qu'elles soient chargées (à la façon des images). Une fois le chargement depuis le Web opéré, la vraie police remplace le rendu temporaire sans qu'il soit nécessaire de recharger la page.

Les fonds

La propriété `background-break` permet de faire courir une image de fond sur toute une boîte, éventuellement des boîtes intérieures, ou de répéter cette image à l'identique sur des boîtes intérieures. Il est possible de mettre comme fond plusieurs images dans la même boîte.

Les propriétés `background-clip` et `background-origin` donne le point de départ de l'image. Cette dernière peut être étendue sur toute la page si elle est plus petite que la page en question, ou étendue jusqu'aux bordures avec un effet de relief, ou répétée avec un espace séparateur modulable.

8.3 Conclusion

Ces recommandations du W3C sont de plus en plus appliquées.

À l'heure actuelle (février 2009), un certain nombre de navigateurs sont compatibles CSS3 et obtiennent un résultat satisfaisant au test Acid3 (`http://acid3.acidtests.org/`) :

- Firefox 3.2a1 (Gecko/20090119) : 93/100

- Opera 10.0 alpha 1 : 100/100
- Safari 4 Developper Preview : 100/100
- Konqueror snapshot r816030 : 73/100
- WebKit (moteur JavaScript V8 de Google Chrome) : 100/100
- Internet Explorer 8 RC1 : 21/100

Microsoft freine l'essor du CSS3 à dessein mais cette stratégie pourrait se retourner contre eux dans la mesure où les CSS3 vont se répandre sur le web. Si de gros sites adoptent la CSS3 en demandant à leurs visiteurs de télécharger Opera ou Firefox dont l'utilisation monte — on le voit dans les statistiques des sites —, alors Microsoft sera obligé d'adopter les recommandations de W3C.

Les développeurs ont intérêt à insérer du CSS3 peu à peu sur leurs sites pour être prêts au moment venu. Il est toujours possible de créer deux feuilles de styles et d'orienter les visiteurs sur l'une ou l'autre selon la version de leur navigateur mais cela oblige à développer deux fois, merci Microsoft !

9

Les CSS se marient bien

Les CSS se marient bien avec le XML, le JavaScript, AJAX (*Asynchronous JavaScript And XML*, ou XML et JavaScript asynchrones), les langages dynamiques de script comme PHP ou ASP. Ces langages compensent les manques des CSS ou les incompatibilités des différents navigateurs, ou encore renforcent et multiplient les possibilités des CSS. Internet marie bien les langages différents sur une même page, même si parfois ce code hétérogène est plus difficile à maintenir ou même à relire.

Ce chapitre va d'abord présenter ces différents langages, afin que vous puissiez en assimiler quelques notions, puis différentes manières de compenser les manques des CSS ou d'en améliorer les fonctionnalités. Parfois ce sont d'ailleurs les CSS qui améliorent ces langages !

Plus le développeur utilise de langages différents, plus il a intérêt à bien documenter son code, en particulier sur la page elle-même, avec des commentaires, des lignes de séparation, des lignes blanches.

Le langage JavaScript est le plus souvent utilisé car c'est le plus souple puisqu'il est exécuté par le navigateur lui-même. Le navigateur possède un moteur JavaScript qui fait tourner le script détecté dans le langage source. Ce processus évite de recharger la page pour opérer un changement. AJAX l'exploite au mieux en le mariant aussi au XML.

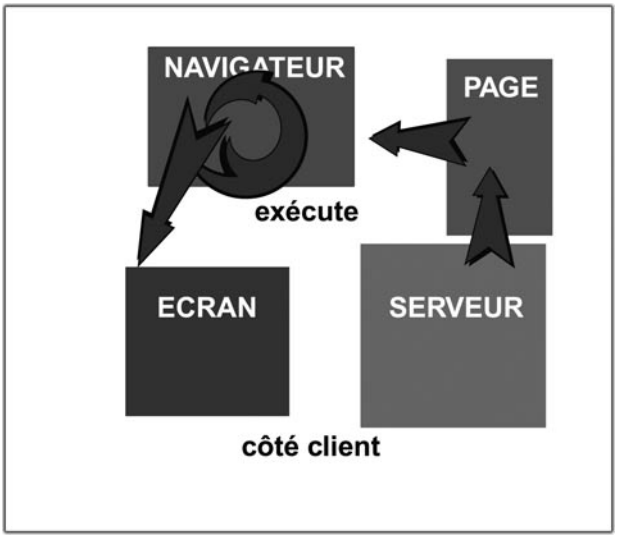
9.1 Le JavaScript

Le JavaScript est un langage de script côté client principalement, c'est-à-dire que le code est exécuté par le client, le navigateur lui-même, en temps réel. Vous pouvez y accéder en affichant le source de la page au moyen de votre navigateur. Le JavaScript se place au milieu du code HTML entre les balises `<SCRIPT>` et `</SCRIPT>`. Un certain nombre de langages de script sont exécutés par le serveur, comme le PHP, l'ASP, etc.

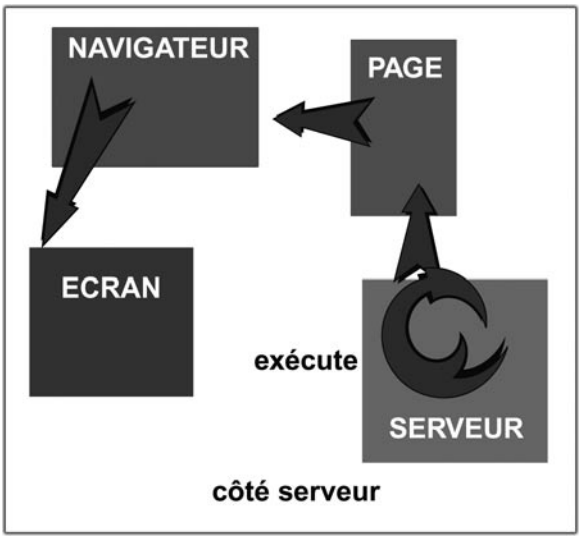
▲ Figure 9.1 : *PHP*

Côté client, côté serveur

Imaginez que le serveur dans un café prépare vos consommations et charge votre commande sur un plateau. Vous, en tant que client, vous prenez votre café et vous ajoutez un sucre. Vous n'êtes pas obligé de mettre le sucre mais ce sucre est là au besoin. Le serveur web va exécuter les commandes qui lui sont destinées (code PHP, ASP…) puis présenter la page web demandée sur un navigateur. Le client est le navigateur web qui affiche une page web placée sur le serveur. Certaines lignes de code (HTML) lui donnent les directives pour présenter la page. Ainsi les liens seront soulignés, les images placées à tel ou tel endroit. En plus du HTML, la page sera enrichie de lignes de code JavaScript. Ce code sera chargé en mémoire vive en attente d'un événement qui déclenchera son exécution (le passage de la souris, un clic, une touche enfoncée…).

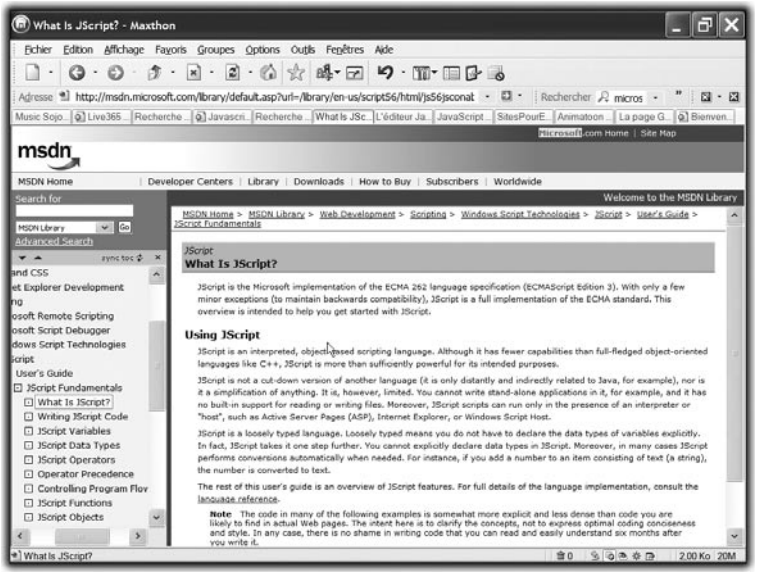


▲ Figure 9.2 : *Processus du langage client*



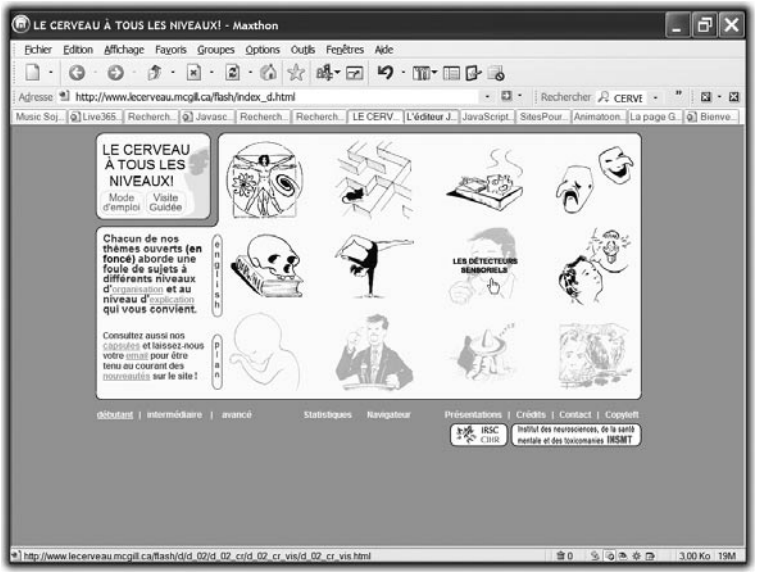
▲ Figure 9.3 : *Processus du langage serveur*

Le JavaScript peut se combiner avec de nombreux autres langages sur la même page avec cette spécificité qu'il est exécuté par le navigateur. Cette spécificité lui donne l'avantage de pouvoir réagir immédiatement à une action du visiteur sans qu'il soit nécessaire de recharger la page. L'inconvénient est que les fonctions JavaScript sont dépendantes du navigateur et de sa version. Le JavaScript a été inventé sur le navigateur Netscape. Puis Microsoft a voulu disposer de son propre JavaScript, en proposant le JScript pour Internet Explorer. Malgré les divergences, les deux navigateurs s'accordent sur un ensemble de fonctions communes. Ainsi il a été proposé par l'organisme de régulation ECMA plusieurs versions du JavaScript.



▲ Figure 9.4 : *Jscript*

JavaScript, le cerveau du navigateur



▲ Figure 9.5 : *Le cerveau du navigateur*

Imaginez le navigateur comme un corps humain. Le HTML composerait les cellules et le code JavaScript serait le cerveau, un lieu où convergent toutes les informations et qui réagit en temps réel. Imaginez que vous vous brûliez le doigt. Vous l'enlevez immédiatement de l'objet brûlant. Le langage JavaScript de la même façon réagit à des événements qui surviennent dans le document ou le navigateur. Il contrôle tous les composants du document, son apparence et son contenu, commande le navigateur, intervient dans les formulaires et dialogue avec le visiteur. Le JavaScript est orienté vers la communication humaine :

- Il a les yeux par-dessus l'épaule du visiteur et vérifie qu'il ne fait pas d'erreur dans un remplissage de formulaire (vérification de formulaires).
- Il met en valeur un texte en mouvement pour être sûr que le regard du visiteur soit accroché par un événement important (défilement de texte).

- Il élabore un menu en arborescence qui s'escamote et se déplie selon le désir (menu dynamique).

▲ Figure 9.6 : *Un menu dynamique*

- Il modifie l'aspect de la page à la période des fêtes de fin d'année.
- Il se souvient du dialogue échangé lors de la dernière visite grâce à un "tiroir" dans le disque dur du visiteur (cookie).

Si vous avez anticipé une action du visiteur par une commande JavaScript, votre page répond par une réaction. Désormais le visiteur attend une réponse immédiate. S'il clique, il ne tolère plus que l'écran soit sourd et aveugle. Dans ce monde virtuel, le JavaScript est devenu indispensable pour engager un dialogue avec les visiteurs. Même si votre site est modeste, il est créé pour communiquer. Ce n'est pas la peine de mettre de la poudre aux yeux avec une multitude d'effets, quelques lignes de code pour souhaiter la bienvenue peuvent suffire.

Quelques amuse-gueules

Commençons par quelques amuse-gueules pour nous ouvrir l'appétit et comprendre les notions de base de manière amusante.

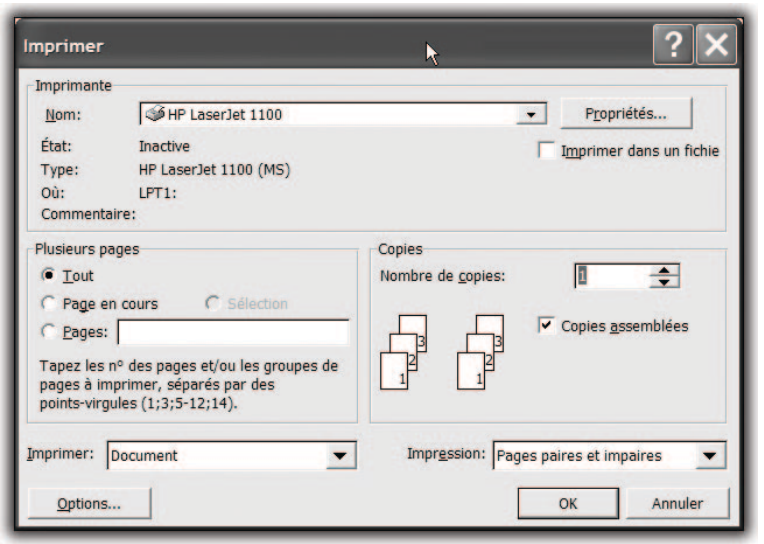
Pour le JavaScript, le document est une armoire contenant une ou plusieurs portes, une ou plusieurs étagères, un ou plusieurs tiroirs, une penderie ou pas, etc. Ainsi l'objet `document` contient une ou plusieurs fenêtres ouvertes qui correspondent à un ou plusieurs objets `window`, un ou plusieurs formulaires, un ou plusieurs liens, etc.

Imprimer

Dans votre document HTML, vous pouvez créer un lien pour imprimer, entre les balises HTML <BODY> et </BODY>.

```
<a href="#" onClick="window.print()">
   imprimez la page</a>
```

Si le navigateur a une version égale ou supérieur à Netscape 4 et Internet Explorer 5, le gestionnaire d'impression sera appelé pour imprimer la page.



▲ Figure 9.7 : *Imprimer*

Le caractère dièse (#) dans le lien fait référence à la page elle-même et permet d'activer la méthode `print()` de l'objet `window`. Le terme `onClick` désigne l'événement provoqué par un clic sur le lien.

Le caractère dièse (#)

Le caractère dièse est utilisé en HTML pour un lien interne de la page (`<a href="#bas de la page">`) afin de désigner une ancre `<a name="bas de la page">`. Dans le cas du JavaScript, comme le dièse est seul, il désigne la page elle-même. Notez que le lien sans objet peut, par exemple, avoir son adresse remplacée aussi par `javascript:void(0);`, qui lance une fonction sans recharger la page.

Retourner en arrière

De la même façon que précédemment, créez un nouveau lien. Ce lien servira à retourner sur la page précédente. Ce bouton imite la commande du navigateur qui permet de revenir en arrière.

```
<a href="#" onClick="history.back()">
   revenir à la page précédente</a>
```

Pour revenir à une page précédente, vous disposez de la méthode `back()` de l'objet `history`, mais vous avez aussi la méthode `go()`, qui propose un équivalent.

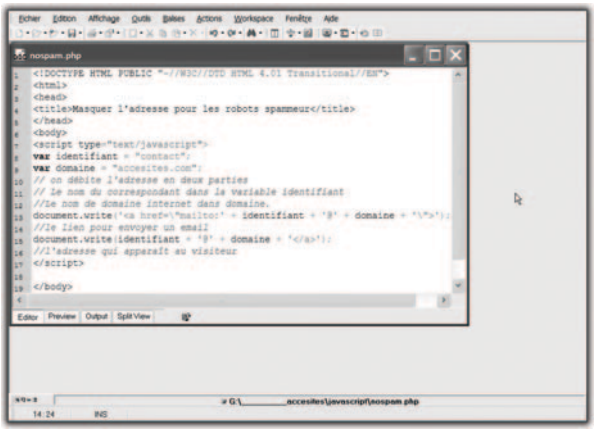
```
<a href="#" onClick="history.go(-1)">
   revenir à la page précédente</a>
```

Pour recharger la page présente, utilisez la méthode `history.go(0)`. Pour aller deux pages plus loin, employez `history.go(2)`.

Tromper les robots

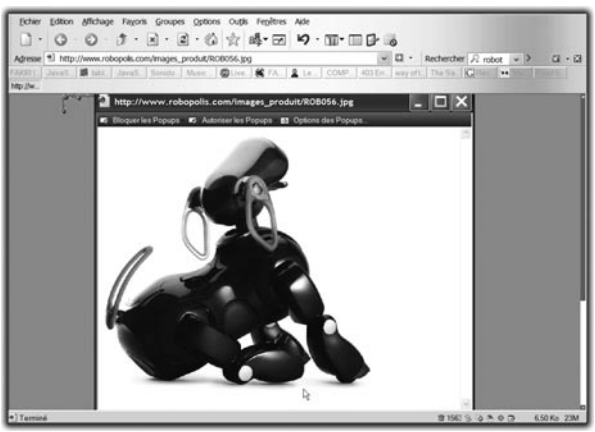
Vous avez sans doute fait l'expérience de recevoir un flot de courriels, principalement en anglais, envahissant votre boîte d'e-mails. Pour créer ces messages, appelés pourriels en français ou spam en anglais, il faut utiliser des robots qui récoltent les adresses sur les sites.

Pour tromper ces robots, vous allez modifier l'apparence du code HTML à l'aide du JavaScript. Vous pourrez insérer ce code n'importe où dans votre page, entre les balises HTML <BODY> et </BODY>.



▲ Figure 9.8 : *Crypter les adresses email*

Ce script est inséré entre deux balises <script>. Ainsi le navigateur identifiera le code JavaScript à exécuter parmi les balises HTML.



▲ Figure 9.9 : *Robots & Co*

Commencez par créer deux variables en les déclarant accompagnées du mot-clé `var` et en leur affectant une valeur via l'opérateur `=`. Suivent deux lignes commençant chacune par deux lignes obliques. Ces deux lignes sont des commentaires, c'est-à-dire des explications sur le code. Le processeur JavaScript contenu dans le navigateur n'exécute pas ces lignes.

Confusion entre l'opérateur d'affectation et l'opérateur d'égalité

Il peut exister une ambiguïté entre l'opérateur d'affectation (=), qui attribue une valeur à une variable, et l'opérateur d'égalité (==), qui teste l'égalité entre deux valeurs ou entre une variable et une valeur.

Suivent deux lignes utilisant la méthode `write()` de l'objet `document`. Cette méthode permet tout simplement d'écrire du texte dans un document. Au lieu d'écrire du texte littéral, le programme mêle du texte et des variables. Le texte est encadré d'apostrophes, tandis que les variables ne demandent pas tant de précautions. Pour concaténer (ou coller) les variables et le texte, JavaScript utilise l'opérateur `+`, qui n'est plus ici le symbole de l'addition. JavaScript détermine le rôle de `+` en analysant le contexte : c'est un opérateur d'addition quand les membres de chaque côté sont des nombres.

```
var somme = 2+3;
```

Vous pouvez aussi aller sur un site qui créera du texte JavaScript aléatoire pour vos adresses e-mail, comme `http://aspirine.org/emailcode.php`. Ne laissez jamais une adresse e-mail nue sur une plage web !

Salutations

Pour que votre site soit un peu plus accueillant, ce petit programme affiche le message "Bonjour!" au chargement de la page. Son exécution repose sur l'événement contenu dans la balise `<body>` : `onLoad`.

```
<html>
<head>
<script>
function salutations() {
alert("Bonjour!");
```

```
}
</script>
</head>
<body onLoad="salutations()">
</body>
</html>
```

Le programme conserve quelques instructions dans une boîte appelée "fonction", logée dans l'en-tête de la page entre les balises <head> et </head>, et encadrée par les balises <script> et </script>.

Une fonction est déclarée par le mot-clé `function` et un nom lui est attribué : ici `salutations`. Les parenthèses qui suivent contiennent 0, 1 ou plusieurs arguments, qui seront traités par les instructions internes de la fonction. Dans ce programme, aucun argument n'est nécessaire. Conservez néanmoins les parenthèses. Les instructions de la fonction se trouvent entre accolades et chaque ligne se termine par un point-virgule (`;`). En fait, le point-virgule termine une instruction. Si une ligne contient plusieurs instructions, elles sont séparées par un point-virgule.

Le point-virgule

Dans un programme JavaScript, il n'est pas nécessaire de terminer une ligne contenant une instruction par un point-virgule car si le point-virgule est absent, il sera implicite et le processeur JavaScript l'ajoutera. Ainsi il faut éviter de couper une instruction en la faisant chevaucher deux lignes car sinon, le programme ajoutera un point-virgule à la fin de chaque ligne et l'instruction sera invalide. En revanche, le point-virgule est obligatoire quand plusieurs instructions sont disposées sur une seule ligne. Il est conseillé, néanmoins, d'ajouter un point-virgule à toutes les fins de ligne, excepté si elles sont terminées par une accolade.

La méthode `alert()` appartient à l'objet `window`. Elle affiche une petite fenêtre ornée d'un bouton OK et, quand l'utilisateur clique dessus, elle ferme la fenêtre.

Récapitulation

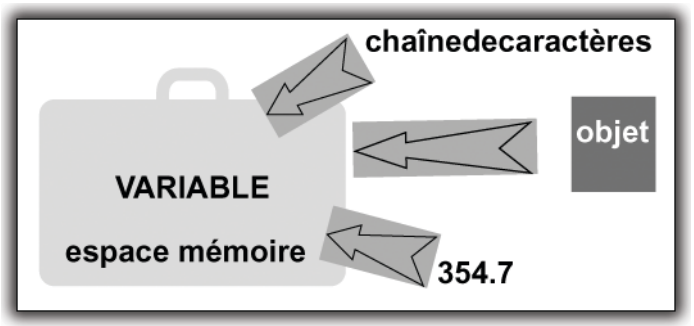
Nous venons de voir qu'en JavaScript, il existe des objets, des méthodes, des événements, des opérateurs, des variables et des fonctions.

La variable, récipient d'une valeur

Pour qu'une variable existe, il faut nécessairement qu'elle ait été déclarée, sinon votre page affichera une erreur JavaScript.

La variable

La variable alloue une portion de la mémoire de votre ordinateur à une donnée et le nom de la variable permet de désigner le contenu de cet emplacement mémoire. Une variable typée permet de bien gérer la mémoire car un objet `Date`, par exemple, prend moins de place que son équivalent en chaîne de caractères.



▲ Figure 9.10 : *Variables*

La déclaration se fait soit d'une manière simple :

```
var a;
```

soit combinée avec une affectation de valeur :

```
var a=10;
```

En JavaScript, les variables sont peu typées. La valeurs peuvent être de type numérique (10), littéral (`"dix"`), booléen (`true`), etc.

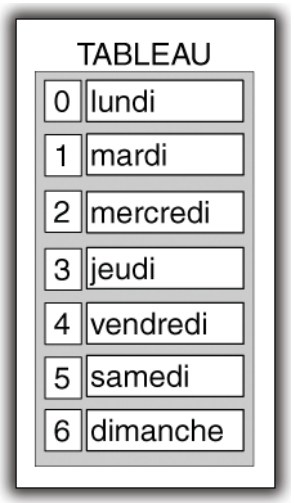
La variable est une sorte de portemanteau : au début du programme, elle porte un manteau puis au milieu un blouson et à la fin un imperméable.

Elle peut être déclarée avec une valeur numérique, comme 10, et ensuite être affectée d'une chaîne de caractères, comme `"dix"`, sans qu'il y ait d'erreur.

Une variable peut contenir plusieurs valeurs. Dans ce cas, on parle de tableau.

Le tableau, placard pour ranger plusieurs données

Un tableau est une collection de valeurs hétérogènes. Elles ont identifiées par un indice. La première case a un indice 0, la deuxième un indice 1, etc. Le tableau est un moyen pratique de stocker plusieurs valeurs ou, à l'intérieur d'une fonction, de retourner un ensemble de valeurs.



◄ Figure 9.11 : *Tableau*

Le tableau peut être créé de différentes manières :

```
var touslesmois= new Array();
```

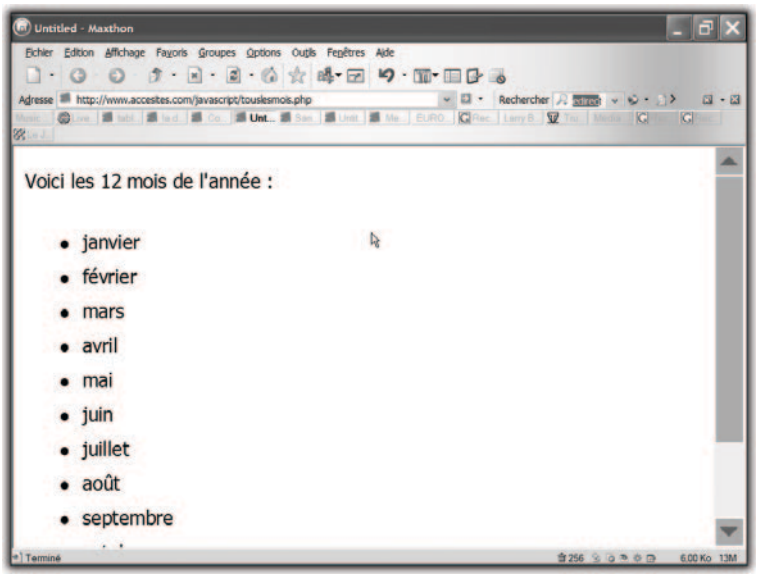
Ce tableau est vide, mais il est possible de lui affecter directement des valeurs de la façon suivante :

```
var touslesmois= new Array("", "janvier",
   "février", "mars", "avril", "mai", "juin",
   "juillet", "août", "septembre", "octobre",
   "novembre", "décembre");
```

Le tableau contient désormais une case vide et les douze mois de l'année en français. Vous allez comprendre assez vite pourquoi la première case est vide.

Pour avoir la valeur du cinquième mois, il faut écrire :

```
document.write("Quel est le nom du cinquième mois ?"
   +touslesmois[5]);
```



▲ Figure 9.12 : *Affichage des mois*

Pour afficher tous les mois, il est nécessaire d'utiliser des instructions de boucle de type `for` :

```
document.write("Voici les 12 mois de l'année: <UL>")
for (i=1 ; i<13 ; i++) {
document.write("<LI>", touslesmois[i] )
```

```
}
document.write("</UL>");
```

En JavaScript, la méthode `getMonth()` permet d'obtenir le mois à partir d'un objet `Date`.

La méthode getMonth()

L'information que renvoie la méthode `getMonth()` est l'indice du mois. Cet indice est `0` pour janvier, et non `1`. Ainsi, le tableau des mois ne nécessiterait plus de première cellule vide.

Les instructions, rouages et moulinettes

Dans l'exemple précédent, une instruction évite de répéter le même code douze fois pour afficher le nom du mois. Elle forme une boucle qui exécute la même instruction douze fois pour afficher la liste des mois.

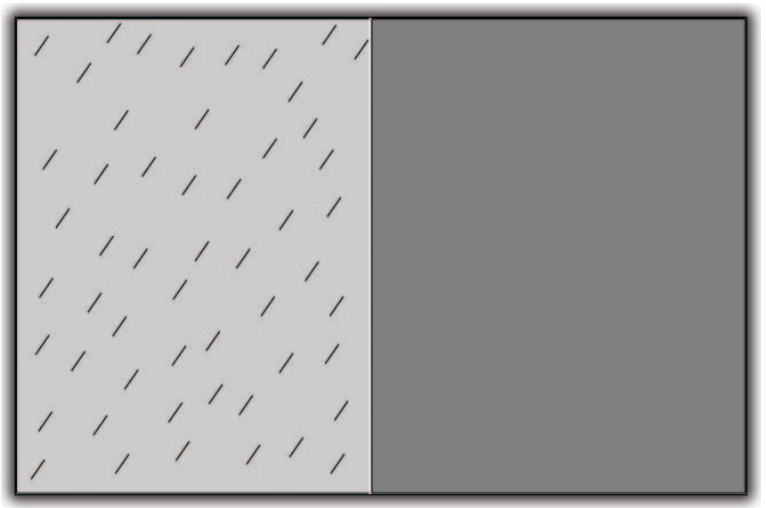
La boucle `for` comprend trois arguments séparés par un point-virgule. Le premier donne la valeur de départ de la boucle. En l'occurrence, ici la variable `i` prend la valeur 1 au premier tour. Le deuxième argument est la valeur limite de `i`. Elle déclenche, une fois atteinte, la sortie de la boucle. Enfin, le dernier argument indique le pas de l'incrémentation de `i` à chaque tour, ici 1.

if, else, else if

Pour tester si un prédicat est vrai ou faux, vous disposez de trois outils : `if`, `else` et `else if`, qui sont complémentaires.

Si {*le ciel est bleu*} alors {*le temps est beau*} sinon {*le temps est nuageux*}.

```
if (ciel=="bleu"){
   temps="beau";
}
else {
   temps="nuageux";
}
```



▲ Figure 9.13 : *Beau temps ou pluie ?*

Cette séquence procède d'une logique. Cela paraît évident, mais des erreurs peuvent se produire :

- si l'on confond = (égalité) et == (affectation d'une valeur) ;
- si la séquence contient une erreur de logique, par exemple :

```
if (ciel!="bleu"){
   temps="pluvieux";
}
else {
   temps="beau";
}
```

Ici le prédicat est faux, non pas à cause de la syntaxe mais de la logique.

L'opérateur !=

L'opérateur != permet un test sur le prédicat "*est différent de*".

Parfois il est intéressant de dévider les éléments d'un tableau ou de les provisionner avec une valeur qui s'incrémente. C'est là que les instructions `while` et `for` deviennent indispensables.

while

`while` est une boucle qui se poursuit tant que le prédicat est vrai :

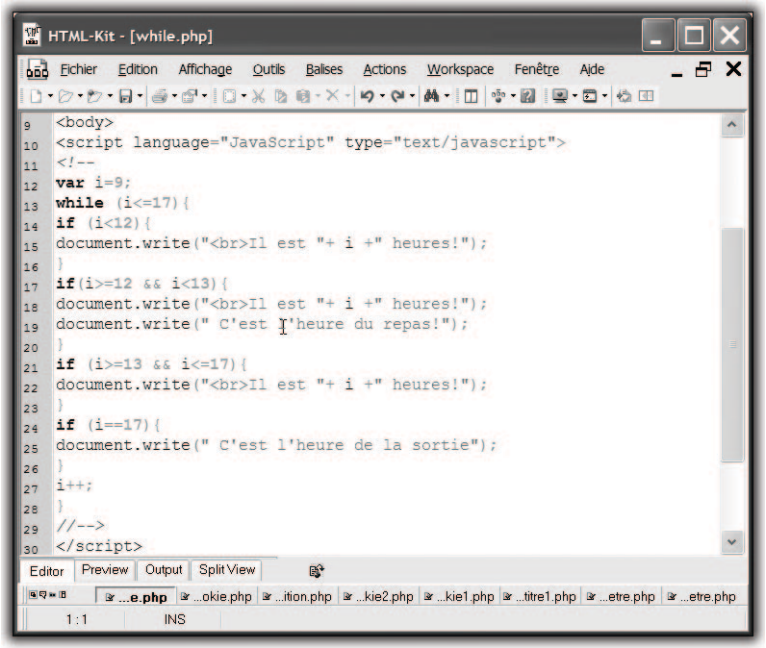
```
<script language="JavaScript"
   type="text/javascript">
<!--
var i=9;
while (i<=17){
  if (i<12){
    document.write("<br>Il est "+ i +" heures!");
  }
  if(i>=12 && i<13){
    document.write("<br>Il est "+ i +" heures!");
    document.write(" C'est l'heure du repas!");
  }
  if (i>=14 && i<=17){
    document.write("<br>Il est "+ i +" heures!");
  }
  if (i==17){
    document.write(" C'est l'heure de la sortie");
  }
  i++;
}
//-->
</script>
```

Une boucle est caractérisée par un incrément, ici `i++`, et un argument de sortie, `while (i<=17)`. Le test de la boucle est le suivant : la boucle se poursuit tant que `i` est plus petit ou égal à 17.

L'opérateur unaire `++` ou `--` permet d'incrémenter ou de décrémenter la valeur d'une variable. Il équivaut à `i=i+1` ou à `i=i-1`.

Si cet opérateur d'incrémentation ou de décrémentation n'est pas présent, la boucle reste sur la première valeur et devient infinie, ce qui peut être dangereux pour une machine. Heureusement, JavaScript a un parapet pour arrêter les boucles infinies et le navigateur, assez vite, propose d'interrompre l'exécution du programme. Si vous remplaciez les `if` par des

`while`, la première boucle à l'intérieur de la grande boucle serait infinie, bloquée sur la première valeur, 9.



▲ Figure 9.14 : *While*

break et continue

Dans une boucle, pour économiser du temps et de la ressource serveur, il est courant d'utiliser `break` dans un test conditionnel. Cette instruction rompt la boucle et évite plusieurs tours inutiles. Tandis que `continue` permet de sauter une boucle sur une condition et de poursuivre à la boucle suivante directement.

for

La boucle `for` est une boucle `while` qui intègre tous les paramètres, c'est-à-dire le début de la boucle, la fin de la boucle et l'incrémentation.:

```
<script language="JavaScript"
   type="text/javascript">
<!--
for (i=9;i<=17;i++){
  if (i<12){
    document.write("<br>Il est "+ i +" heures!");
  }
  if(i>=12 && i<13){
    document.write("<br>Il est "+ i +" heures!");
    document.write(" C'est l'heure du repas!");
  }
  if (i>=13 && i<=17){
    document.write("<br>Il est "+ i +" heures!");
  }
  if (i==17){
    document.write(" C'est l'heure de la sortie");
  }
}
//-->
</script>
```

Les arguments sont séparés par un point-virgule :

- la valeur d'initialisation ;
- la valeur limite ;
- l'incrément ;

Plusieurs variables

Il est possible d'insérer plusieurs variables dans les arguments de `for` : il suffit pour cela de les séparer par une virgule, par exemple `for (i=9,j=0;i<=17;i++,j+=3)`.

Préférez les boucles `for` aux boucles `while`, excepté quand vous avez déjà la valeur d'initialisation.

Les fonctions, automates à votre service

La fonction est un petit programme autonome que l'on retrouve principalement dans les en-têtes de page. Voici les caractéristiques de la fonction :

- mot-clé `function` ;
- nom ;
- arguments ou absence d'arguments entre parenthèses ;
- instructions entre accolades ;
- valeur retournée ou non avec le mot-clé `return`.

La fonction

La fonction est un ensemble nommé d'instructions entre accolades, dans laquelle on peut injecter des arguments sous forme de valeurs entre parenthèses et qui peut retourner une valeur par l'intermédiaire du mot-clé `return`.

La fonction est appelée par un événement, comme `onClick`, `onSubmit`, etc.

Dès que le programme rencontre un `return`, la fonction s'arrête. Si vous subordonnez la soumission d'un formulaire à une fonction et qu'elle retourne `false`, l'action avorte et le formulaire n'est pas envoyé vers sa destination, jusqu'à ce que les conditions d'envoi soient réunies et que la fonction retourne `true`.

```
<form action="traitement.php" method="post" onSubmit="return
✂< verif_email()">
```

Les objets, outils spécialisés

Les objets sont des containers contenant un ensemble de propriétés dotées chacune d'un nom et d'une valeur. Les propriétés sont donc des sortes de variables inhérentes à l'objet.

Par exemple, `screen.height` définit la propriété `height` (hauteur) de l'objet `screen` (écran). Cette variable, que nous allons appeler "pro-

priété", prend une valeur implicite dès l'ouverture de la page selon la résolution de l'écran du visiteur du site.

L'objet et ses instances

L'instance d'un objet est créée avec la méthode `new()` suivie du nom de l'objet. L'objet modèle est le prototype de l'objet et son nom commence par une capitale. L'instance a les mêmes attributs que l'objet lui-même et permet d'effectuer des transformations. Tandis que des modifications sont faites sur l'instance, l'objet lui-même demeure vierge.

Le point qui sépare `screen` et `height` signifie que la propriété `height` appartient à `screen`.

L'objet `screen` est un objet natif contenu dans le noyau de programme JavaScript. Ainsi, il n'est pas nécessaire de le créer.

L'objet contient aussi des méthodes, qui sont les fonctions intégrées de l'objet. Prenons un autre objet : `Date`. Vous auriez peut-être envie de mettre la date du jour sur votre site ? La date affichée par un JavaScript est celle de l'ordinateur de l'utilisateur. Ainsi deux visiteurs, l'un au Québec, l'autre en France, auront deux dates différentes tandis qu'ils sont sur la même page au même moment.

1. Pour créer un objet `Date`, utilisez la méthode `new()`.

```
madate= new Date();
```

2. Pour la mettre au format local sans faire trop de manipulations, utilisez la méthode `toLocaleString()`. Ici la méthode `toLocaleString()` formate la date selon le pays du visiteur.

```
datelocale=ladate.toLocaleString();
document.write("Nous sommes le " + datelocale);
```

Les objets `String`, `Math` ou `Date` sont natifs, c'est-à-dire qu'ils sont déjà dans le noyau JavaScript propre à chaque navigateur. Mais vous pouvez construire un objet en utilisant un constructeur. Il s'agit d'une fonction qui crée un objet à la demande.

Par exemple, si vous en avez assez de formater les noms recueillis dans un formulaire pour vous adresser au visiteur, créez d'abord le constructeur de l'objet `VotreNom` :

```
function VotreNom(prenom,nom) {
  this.prenom = prenom;
  this.nom = nom;
}
```

Le mot-clé `this` représente l'objet courant. Ainsi, les arguments envoyés à la fonction deviennent respectivement les propriétés `prenom` et `nom` de l'objet `VotreNom`.

Faites alors appel au constructeur avec le contenu des champs `prenom` et `nom` du formulaire :

```
votrenom=new VotreNom(
   document.forms[0].prenom.value,
   document.forms[0].nom.value);
```

Une fois le formulaire rempli, remerciez le visiteur :

```
alert (votrenom.prenom + " " + votrenom.nom);
```

Il vous faudra commencer le prénom et le nom par une capitale et vous aurez donc besoin d'une fonction.

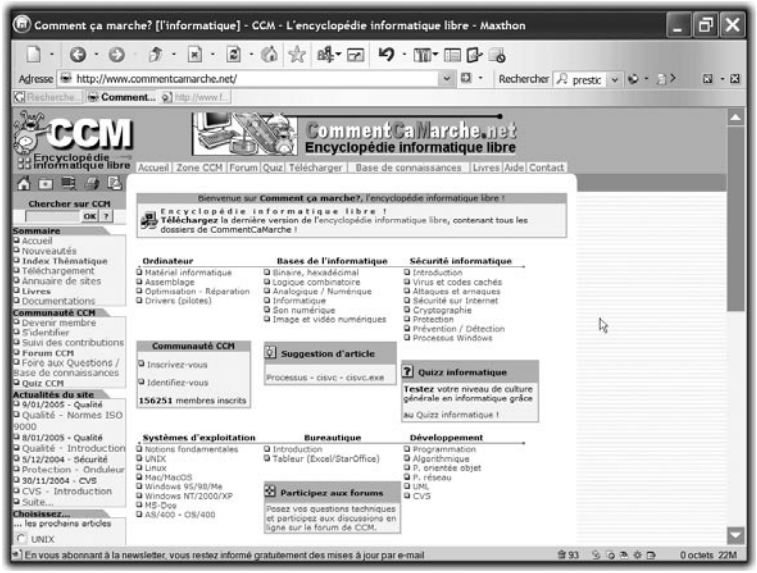
Les différents navigateurs et le JavaScript

Pour l'instant, le navigateur le plus compatible JavaScript est Internet Explorer. L'acclimatation des navigateurs à JavaScript dépend des différentes versions des navigateurs et du JavaScript.

▲ Figure 9.15 : *Avec Netscape*

Comment tout cela a commencé ?

Netscape a commencé par développer dans les années 90 un langage de script, LiveScript, qui agissait directement sur le client web (le navigateur). En 1995, Sun Microsystems récupère LiveScript et l'appelle JavaScript, pour ajouter encore un peu de notoriété au langage que Sun développait alors, Java. JavaScript est pris en charge au fil de son évolution par différentes versions de Netscape. Microsoft, intéressé, décide alors de l'intégrer à Internet Explorer. Microsoft ne se contente pas d'intégrer JavaScript mais crée un langage dérivé, le JScript, qui ressemble à JavaScript sans être du JavaScript. JScript a l'inconvénient de ne s'exécuter que sous Internet Explorer sans apporter d'améliorations intéressantes par rapport à JavaScript (voir Figure 9.16).



▲ Figure 9.16 : *Avec Internet Explorer*

Java et JavaScript

Le JavaScript n'est pas du Java, comme son nom le laisse supposer. Ils ont quelques vagues parentés, comme l'utilisation des objets, mais leur ressemblance s'arrête là.

Des navigateurs qui ne naviguent pas dans les mêmes eaux !

Tab. 9.1 : L'intégration des différentes versions de JavaScript dans les versions successives des navigateurs

Version de JavaScript	Navigateur
JavaScript 1.0	Netscape Navigator 2.0, Internet Explorer 3.0, Opera, Mozilla

Tab. 9.1 : L'intégration des différentes versions de JavaScript dans les versions successives des navigateurs

Version de JavaScript	Navigateur
JavaScript 1.1	Netscape Navigator 3.0, Opera, Mozilla
JavaScript 1.2	Netscape Navigator 4.0/4.05, Internet Explorer 4.0, Opera, Mozilla
JavaScript 1.3	Netscape Navigator 4.06, Internet Explorer 5.0, Opera, Mozilla
JavaScript 1.4	Netscape Navigator 6.0, Internet Explorer 6.0, Opera, Mozilla
JavaScript 1.5	Netscape Navigator 6.0, Mozilla

Pour que le code n'apparaisse pas en clair dans les navigateurs antiques, le script est systématiquement mis en commentaires.

```
<script language="Javascript">
<!--
Placez ici le code
// -->
</script>
```

Plus le script est sophistiqué, moins le nombre de navigateurs qui le comprennent est grand. Il est donc conseillé de choisir, pour les mêmes effets, le script le plus largement compris.

Le jeu "Chercher la mouche"

Vous allez faire un petit jeu emprunté à l'excellent site `www.biblioscript.com/javascript/`. Il s'appelle "*chercher la mouche*" et consiste à placer une mouche invisible sur l'écran et à la trouver. Les dames de la cour pratiquaient ce jeu dans un but moins avouable, avec un homme aux yeux bandés.

Le langage de programmation sert à recevoir, à transformer et à transmettre de l'information. Le programme joue d'abord une sorte de roue avec des numéros et s'arrête au hasard sur l'un de ces numéros puis tourne pour tirer un deuxième nombre. La combinaison de ces deux nombres donne la position virtuelle de l'objet `mouche`.

Chaque coup du joueur consiste à recueillir une position x et une position y via un formulaire et à les comparer avec la position de la mouche pour indiquer si la proposition du joueur est trop haute ou trop à droite. Le but du jeu est de trouver cette position en portant le moins de coups possible. Vous allez adapter le jeu en fonction de la résolution de l'écran du joueur. Vous pourriez aussi permettre de jouer à deux joueurs, mais tel n'est pas le propos ici.

1. Commencez par recueillir la résolution de l'écran du joueur dans deux variables.

```
hauteur_ecran = screen.height;
largeur_ecran = screen.width;
```

En JavaScript nous avons des objets, des propriétés et des méthodes.

Ces valeurs sont recueillies dans des récipients nommés "*variables*". Ici `hauteur_ecran` recueille la valeur de `screen.height` et `largeur_ecran`, celle de `screen.width`.

La dénomination des variables

Pour nommer une variable, vous ne pouvez employer d'espaces. Son nom peut se composer de lettres de l'alphabet, de chiffres, et des caractères _ et &. Il peut commencer par une lettre ou un blanc souligné (_).

On peut dire que la propriété `height` est une variable à l'intérieur de l'objet `screen`. Ainsi, vous auriez pu écrire :

```
hauteur_ecran = screen["height"];
largeur_ecran = screen["width"];
```

Ici, l'objet est vu comme un tableau associatif. Un tableau est une variable qui contient plusieurs cases, ce qui lui permet de ranger des valeurs différentes dans un même ensemble.

Les valeurs `height` et `width` sont rangées dans une armoire `screen`.

Maintenant, vous allez cacher la mouche, quelque part dans l'écran, de façon aléatoire. Pour cela, vous avez un objet à votre disposition, `Math`, et sa méthode `random()`.

La méthode random()

La méthode `random()` génère un nombre pseudo-aléatoire entre `0.0` et `1.0`. Il faut donc le multiplier par la taille de l'écran pour obtenir un nombre entre 0 et la définition de l'affichage en pixels et savoir ainsi sur quel pixel la mouche s'est posée.

Les lignes précédentes sont un ensemble d'instructions liées à une condition. Le caractère `=` est un opérateur d'affectation de valeur à la variable. Le programme est lu de droite à gauche par la machine, qui reconnaît d'abord la valeur puis la variable à laquelle elle est affectée.

2. Pour positionner la mouche :

```
var valeurx = Math.round(Math.random() * largeur);
var valeury = Math.round(Math.random() * hauteur);
var i=0;
```

Ces valeurs sont entrées au chargement de la page et ne varient pas au fur et à mesure des appels à la fonction.

3. L'instruction `window.prompt()` ouvre une fenêtre dans laquelle l'utilisateur peut entrer un texte.

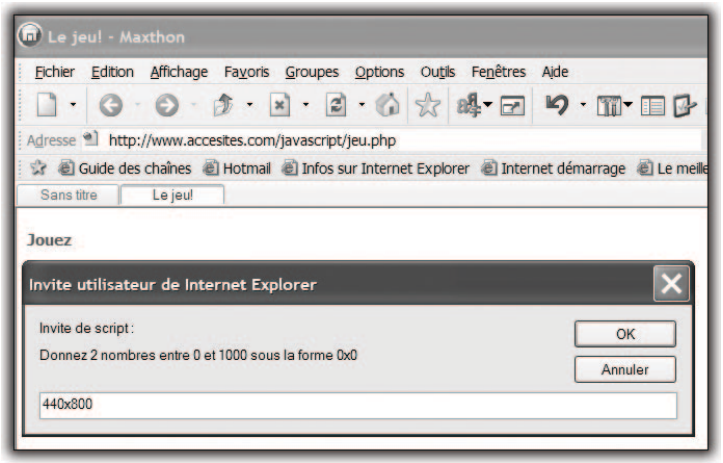
4. Recueillez ce texte dans la variable `reponse`.

5. Demandez à l'utilisateur d'entrer les coordonnées de la mouche sous la forme `0x0`.

6. Traitez ensuite le texte pour récupérer les valeurs qui encadrent `x`.

7. Utilisez la méthode `split()`, qui envoie les valeurs dans un tableau nommé `coordonnees`.

```
function question() {
  i++;
  var reponse=window.prompt("Donnez 2 nombres entre 0 et "
   + multiplicateur + " sous la forme 0x0","");
  coordonnees=reponse.split("x");
```



▲ Figure 9.17 : *Le jeu*

8. Transformez la valeur recueillie en nombres par la méthode `parseInt()`.

```
largeur=parseInt(coordonnees[0]);
hauteur=parseInt(coordonnees[1]);
```

La méthode parseInt()

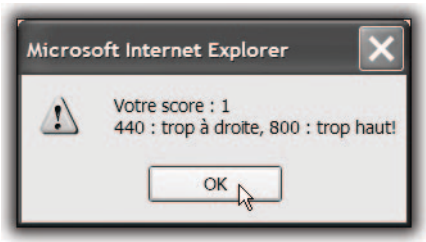
Le texte conservé dans le tableau est littéral. Pour pouvoir devenir un nombre, il devra subir le traitement de la méthode `parseInt()`. Il en est de même pour toute valeur récupérée sur un formulaire ou une fenêtre `prompt()`. En cas d'oubli, l'opérateur ne serait plus un opérateur d'addition mais de concaténation. Ainsi `1+1` deviendrait `11`, et non pas `2`.

9. La variable `avertissement` accueille le texte littéral qui sera affiché après chaque tentative du joueur.

10. L'opérateur permet d'incrémenter une nouvelle chaîne dans le message final, selon la position proposée, à l'issue d'une série de tests sur les valeurs `largeur` et `hauteur`, qui sont comparées à la position `valeurx` et `valeury`.

11. Le mot-clé `return` sert seulement à stopper la fonction car aucune action n'est subordonnée au succès de cette fonction.

```
  var avertissement="Votre score : " + i + "\n";
  if (largeur==valeurx && hauteur==valeury){
    avertissement+= "Bravo, vous avez gagné!";
    return true;
  }
  else {
    avertissement+= largeur + " : ";
    if (largeur<valeurx){
    avertissement+= "trop à gauche, ";
  }
  if (largeur>valeurx){
    avertissement+= "trop à droite, ";
  }
  if (largeur==valeurx){
    avertissement+= "Les coordonnées hozizontales
   sont correctes, ";
  }
  avertissement+= hauteur + " : ";
  if (hauteur==valeury){
    avertissement+= "les coordonnées verticales
   sont correctes";
  }
  if (hauteur>valeury){
    avertissement+= "trop haut!";
  }
  if (hauteur<valeury){
    avertissement+= "trop bas!";
  }
}
alert(avertissement);
}
```



◄ Figure 9.18 : *Résultat du jeu*

12. Il vous reste à insérer un événement déclencheur de la fonction. Ici utilisez un lien vide et subordonnez l'appel à la fonction à l'événement `onClick`. Dans le corps de la page, il suffit donc d'ajouter :

```
<a href="#" onClick="question()">Jouez</a>
```

À chaque clic sur le lien, le joueur propose des coordonnées qui sont comparées à celles qui ont été fournies aléatoirement. Un message d'alerte le guide ensuite au fur et à mesure. Le but est de trouver la mouche en cliquant le moins souvent sur le lien. Le score est indiqué à chaque tentative.

Quelques outils complémentaires

Vous avez vu les principales notions de JavaScript dans ce chapitre. Pourtant, il vous manque encore certains outils d'affinage qui permettent de gagner du temps et d'améliorer la lisibilité des programmes. Pour le moment, ne vous en occupez pas trop, vous les retrouverez plus tard dans des exemples qui vous feront voir l'intérêt de les utiliser.

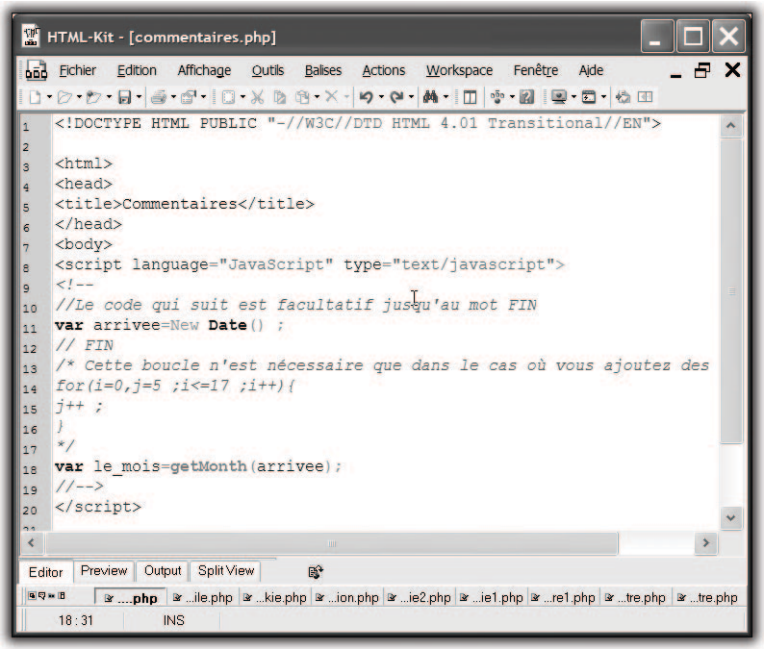
Les commentaires

Les commentaires peuvent venir sur une ligne. Dans ce cas, ils sont précédés de deux barres obliques. Ils peuvent occuper toute une ligne ou un bout de ligne après des instructions JavaScript. Quand ils occupent plusieurs lignes, ils sont ouverts par une barre oblique suivie d'un astérisque et fermés par un astérisque suivi d'une barre oblique.

```
//Le code qui suit est facultatif jusqu'au mot FIN
var arrivee=New Date() ;
// FIN
```

```
/* Cette boucle n'est nécessaire que si
   vous ajoutez des données
for(i=0,j=5 ;i<=17 ;i++){
  j++ ;
}
*/
```

Dans cet exemple, la variable `arrivee` prendra une valeur. Par contre, la boucle `for` ne sera pas exécutée puisqu'elle se situe dans des commentaires.



▲ Figure 9.19 : *Commentaires*

Les commentaires sont utiles si quelqu'un vous aide ou reprend votre site. Ils le seront également pour vous, quand après plusieurs mois, vous reprendrez un script et que vous ne vous souviendrez plus de ce que vous avez fait ni à quoi sert telle ou telle variable. Une petite ligne de commentaires suffit parfois à décrypter un code.

Les commentaires sont précieux au cours d'un débogage. Il permettent de désactiver des parties de code et de se concentrer sur d'autres parties. Si le bogue persiste, cela permet d'écarter les portions de code visées. Les combiner avec des fenêtres d'alerte affichant les valeurs de certaines variables au cours de l'exécution du code permet de localiser la panne.

Le switch ou commutateur

Le `switch` est comme un "commutateur de lignes téléphoniques" : vous lui fournissez une valeur et il énumère une suite de tests sur cette valeur pour la retrouver parmi une série répertoriée, puis exécute une instruction avant de sortir avec la commande `break`, qui rompt l'exécution de la recherche.

Le mot-clé `switch` initialise le bloc avec comme argument une expression. Le mot-clé `case` est suivi d'une valeur puis d'un deux-points et enfin d'une instruction. Le mot-clé `break` évite de poursuivre l'exécution du code dès que la valeur recherchée est trouvée. Un mot-clé facultatif `default` propose une instruction en cas d'échec de la recherche. Cette dernière instruction sera exécutée si la valeur n'est pas trouvée. L'instruction `switch` permet de parcourir rapidement plusieurs possibilités avec une valeur simple comme argument.

```
switch (i) {
  case 1 : var item = new Array('balai',
    'éponge gratounette', 'serpillère'); break;
  case 2 : var item = new Array ('télévision',
   'lecteur DVD','home-cinéma'); break;
  case 3 : var item = new Array ('arrosoir','bêche',
   'semoir'); break;
  default : var item = new Array(); break;
}
```

Ici, la variable `i` a déjà reçue une valeur d'un élément de formulaire. Selon sa valeur, qui va jusqu'à 3, c'est l'un ou l'autre des tableaux qui est désigné.

Un opérateur ternaire plus élégant que le if… else

Cet opérateur rassemble trois parties en une seule ligne et remplace avantageusement `if… else`. Vous allez reprendre l'exemple précédent utilisant `if… else` :

```
if (ciel!="bleu"){
  temps="pluvieux";
}
else {
  temps="beau";
```

Maintenant récrivez-le avec l'opérateur ternaire :

```
var ciel!="bleu" ? temps=pluvieux : temps="beau";
```

L'opérateur ternaire est composé de trois membres

- le test `ciel!="bleu"` suivi d'un point d'interrogation ;
- l'instruction si le test est vrai, suivie d'un deux-points ;
- l'instruction si le test est faux, terminée par un point-virgule.

Comme ce code prend une seule ligne, le script est plus ramassé et donc plus lisible.

L'objet Arguments

L'objet `Arguments` est spécial. Il se comporte comme un tableau mais n'est pas un objet `Array()`. Il contient tous les arguments envoyés lors de l'appel à une fonction.

Les arguments d'une fonction

Les arguments sont les variables nommées entre les parenthèses qui suivent la déclaration du nom de la fonction.

Cet objet est intéressant à plus d'un titre. Ainsi, pour une fonction qui ne reçoit qu'un argument, l'accès à cet argument peut se faire par son nom ou par `arguments[0]`. Si vous envoyez plus d'arguments que prévu dans la fonction, les arguments anonymes seront stockés dans `arguments[1]`, `arguments[2]`, etc. Cela permet de créer des fonctions à géométrie variable, réutilisables dans nombre de situations différentes.

Une propriété lui est adjointe, `length`, qui donne le nombre d'arguments dans l'objet.

```
function somme(a,b){
  if (arguments.length<2){
    var reponse=alert("Entrez deux nombres!");
  }
  else{
    var reponse=a+b;
    for (i=2;i<arguments.length;i++){
      var reponse=reponse+arguments[i];
    }
  }
alert("La somme fait " + reponse);
}
```

Selon le nombre d'arguments envoyés à la fonction, le calcul sera adapté. C'est ce que l'on appelle l'encapsulation, c'est-à-dire que cette fonction pourra être insérée dans n'importe quelle page sans demander aucune modification.

Les règles idéales d'encapsulation sont les suivantes :

- Le nom de la fonction doit être explicite, c'est-à-dire clairement indiquer le rôle de la fonction en question.
- Elle doit accueillir un nombre variable d'arguments selon les besoins.
- Elle doit accueillir tout type de données.

Évidemment, vous devez essayer d'appliquer ces règles dans la mesure du possible. Mais les fonctions ne répondent que rarement à ces trois règles.

Les cookies

Les cookies sont de petits fichiers stockant des variables. Ils peuvent durer le temps de la session de navigation, c'est-à-dire qu'ils sont détruits quand l'utilisateur ferme son navigateur. Cette configuration est celle par défaut, si vous ne spécifiez aucune date d'expiration.

Tab. 9.2 : Les propriétés de l'objet cookie

Nom	Fonction	Valeur par défaut ou statut
`name`	Nom du cookie	Obligatoire

Tab. 9.2 : Les propriétés de l'objet cookie

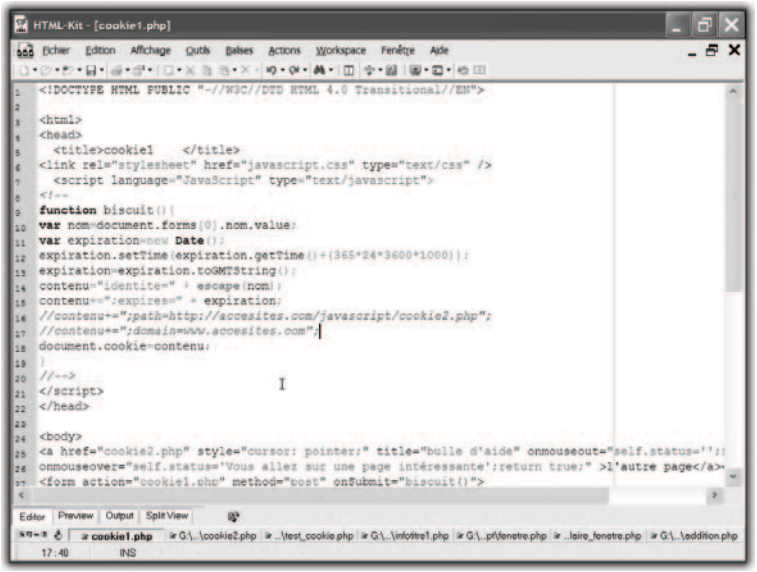
Nom	Fonction	Valeur par défaut ou statut
`expires`	Temps de vie	Par défaut, le temps de l'ouverture du navigateur, au format `Fri, 05-Nov-2004 11:13:59 GMT` (lui affecter une date du passé détruit le cookie)
`path`	Provenance du cookie	Par défaut, le chemin vers la page de provenance
`domain`	Domaine qui l'utilise	Par défaut, le nom du domaine qui l'a créé
`secure`	Accès du cookie uniquement sous *https* (répertoire sécurisé)	

L'usager peut refuser les cookies

L'usager, pour de multiples raisons, peut refuser l'enregistrement de cookies sur son disque dur en réglant les options de son navigateur.

Vous insérez un formulaire récupérant les coordonnées de votre visiteur et vous en profitez pour stocker son nom dans un cookie pour pouvoir l'accueillir de façon personnalisée chaque fois qu'il se connecte à votre site. Pour cela, créez la fonction `biscuit()` comme suit :

```
function biscuit(){
  var nom=document.forms[0].nom.value;
  var expiration=new Date();
  expiration.setTime(expiration.getTime()+
    (365*24*3600*1000));
  expiration=expiration.toGMTString();
  contenu="identite=" + escape(nom);
  contenu+=";expires=" + expiration;
  document.cookie=contenu;
}
```



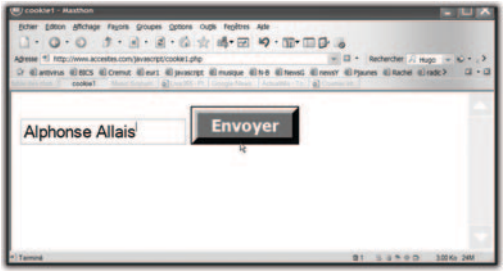
▲ Figure 9.20 : *Fonction biscuit()*

1. La fonction `biscuit()` commence par construire les paramètres qui seront stockés dans le cookie. Le nom est récupéré depuis le champ *Nom* du formulaire. La date d'expiration est repoussée à un an à compter de la date courante. Le temps courant est donné en millisecondes depuis le 1er janvier 1970. À cela s'ajoute les millisecondes qui constituent la durée d'une année. Ce temps brut est ensuite formaté en temps GMT par la méthode `toGMTString()`.

Le 1er janvier 1970

Le temps dans les programmes informatiques est calculé en millisecondes depuis le 1er janvier 1970. Au début des ordinateurs personnels, la mémoire était faible et les programmes se servaient donc de cette date comme repère pour créer une autre date. Or l'ère chrétienne commence le 1er janvier 0000. La méthode `getTime()` calcule cet écart de temps entre le moment courant et ce point de repère.

2. Les paramètres sont ensuite insérés un à un, séparés par un point-virgule, dans la variable `contenu`, qui sera déposée dans l'objet `document.cookie`.



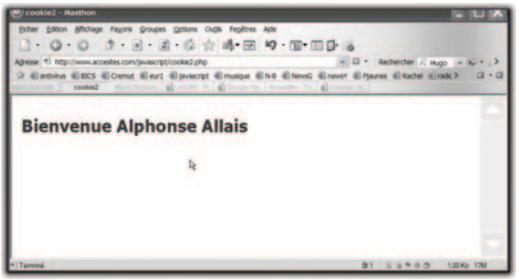
◄ Figure 9.21 : *Cookie 1*

escape()

Vous ne pouvez utiliser le caractère de séparation dans la valeur de la variable, en l'occurrence le point-virgule. Cette interdiction s'est étendue à l'espace et à la virgule. Pour néanmoins pouvoir les intégrer, utilisez la méthode `escape()`. Si vous utilisez cette méthode pour écrire un cookie, il sera nécessaire de lire les valeurs avec la méthode `unescape()`.

Le cookie est créé au moment de la soumission du formulaire par l'événement `OnSubmit="biscuit()"` inclus dans la balise `form`.

Maintenant que cette information est cachée sur le disque dur de votre visiteur, il faut inclure du code pour la récupérer.



▲ Figure 9.22 : *Cookie 2*

3. Dans le body de la page qui affichera le contenu du cookie, insérez par exemple ce code :

```
  <script language="JavaScript" type="text/javascript">
<!--
LesCookies = document.cookie;
// lit le cookie
Debut = LesCookies.indexOf("=");
// relève la position du caractère "="
LongueurCookie = LesCookies.length ;
// longueur du cookie.
Fin = LesCookies.indexOf(";");
// relève la position du caractère ";"
if (Fin == -1) {
// debut du test
Fin = LongueurCookie ;
// la fin est la longueur du cookie,
   //car le caractère ";" n'existe pas
} // fin du test
VotreNom = LesCookies.substring(Debut + 1, Fin);
// extrait le nom
document.write("<b>Bienvenue " +
   unescape(VotreNom) + "</b>");
// affiche le nom dans la page
//-->
</script>
```

Vous utilisez largement les méthodes `indexOf()` et `substring()` pour déceler limites (`;`) et séparateurs (`=`) ou pour engranger les différentes valeurs encadrées par ces signes. N'oubliez pas la méthode `unescape()` pour remettre les espaces éventuels en forme lisible !

4. Les valeurs peuvent aussi être communiquées au programme par un `prompt()`.

```
var nom=window.prompt
   ("Quel sont vos prénom et nom ?","");
contenu="identite=" + escape(nom);
contenu+=";expires=" + expiration;
document.cookie=contenu;
```

5. Ici, il peut être intéressant d'ajouter les fonctions vues précédemment pour formater les noms.

Les cookies sont un moyen pratique pour conserver une d'information sur plusieurs pages du site. L'inconvénient est que certains visiteurs ont choisi de ne pas accepter les cookies. Par ailleurs, la taille des informations est limitée ainsi que le nombre de cookies.

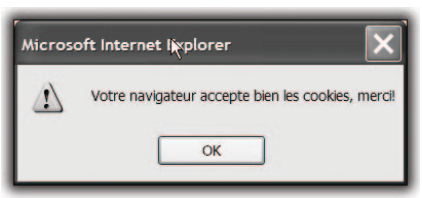
Les limites des cookies

Un ordinateur ne peut contenir que trois cents cookies à la fois, de plus de 4 Ko chacun. La limite est de vingt cookies par serveur maximum.

Tester si les cookies sont acceptés

Comment savoir si le navigateur d'un visiteur accepte les cookies ? Pour cela, créez une page de test sur votre site et demandez à vos visiteurs d'y passer s'ils ne sont pas certains que l'option d'acceptation de cookies est choisie.

```
function test_cookie( ) {
document.cookie = "test= test_cookie";
var pos = document.cookie.indexOf( "test=" );
if( pos == -1 )
{
alert( "Veuillez activer les cookies sur votre
      navigateur, s'il vous plaît!" );
return "non";
}
alert( "Votre navigateur accepte bien les cookies,
        merci!" );
return "oui";
}
```



◄ Figure 9.23 : *Test cookie*

Si le test est passé avec succès, un lien vers la page est affiché avec le message qui donne le résultat du test.

```
if (test_cookie()=="oui"){
document.write('<a href="monautrepage.php">
      mon autre page</a>');
}
```

Un compteur

Voici un autre exemple d'application : un compteur de visiteurs. Le programme est composé de plusieurs fonctions encapsulées.

L'encapsulation

L'encapsulation consiste à écrire un programme de sorte qu'il puisse être utilisé par différents scripts. En JavaScript, c'est une fonction ou un objet. Ici la fonction `ranger_biscuit()` peut être appelée dans n'importe quelle circonstance.

Le compteur vous servira à identifier les visiteurs sur votre site et à reconnaître les fidèles qui reviennent régulièrement. Afficher le nombre de visiteurs sur votre page rangerait votre site dans les antiquités encore en ligne. Par contre, un visiteur sera flatté d'avoir été remarqué comme étant intéressé par le thème du site et, éventuellement, d'avoir un petit privilège pour cette raison.

L'application ici a une architecture savante. Voyez d'abord les lignes de code puis les différentes fonctions au fur et à mesure qu'elles sont appelées :

```
var expiradate = new Date();
var visites;
expdate.setTime(expiradate.getTime() +
   (24 * 60 * 60 * 1000 * 365));
if(!(visites = manger_biscuit("visites"))) {
  visites = 0;
}
else {
  visites++;
}
```

```
ranger_biscuit("visites", visites, expdate, "/",
   null, false);
```

Les deux premières lignes déclarent et initialisent les variables `expiradate` et `visites`.

Ensuite, chaque variable est nourrie. La variable `expiradate` repousse la date d'une année. La variable `visites` est affectée du chiffre `0` ou incrémentée selon le résultat ou non de l'appel à la fonction `manger_biscuit()`. Le test porte sur le résultat de l'appel à la fonction.

Le contenu de la propriété cookie

La propriété `cookie` de l'objet `cookie` contient les paires `nom=valeur` de tous les cookies disponibles pour cette page particulière. En outre, chaque cookie contient des caractéristiques facultatives comme `expires`, la date d'expiration, `path`, le chemin pour lequel il est valide, `domain`, le domaine du serveur, `secure`, le drapeau sécurité (`true` ou `false`). Quand vous allez fouiller dans un cookie grâce à `document.cookie`, vous avez une ou plusieurs paires `nom=valeur`. Le nom est celui du cookie et la valeur, sa valeur. La chaîne obtenue est terminée par un point-virgule. Si la page ne dispose que d'un seul cookie, les paires sont séparées par un point-virgule suivi d'un espace. Pour tester votre cookie, vous pouvez utiliser l'expression `document.write(document.cookie);`. Ainsi vous afficherez son contenu.

Voyez les fonctions. Les deux principales sont appelées dans les lignes de code précédentes : `manger_biscuit()` et `ranger_biscuit()`.

La fonction `manger_biscuit()` récupère les données du cookie. L'argument employé est le nom du cookie. Rappel : `document.cookie` est une chaîne qui peut contenir un ou plusieurs cookies.

La première partie de la fonction s'attache à identifier la longueur de la chaîne qui suit l'expression `nom=` (c'est-à-dire le mot `visites`). Ici la variable qui contient la valeur du cookie prendra le nom `$visites`.

Une première évaluation du cookie valide le fait que la taille du biscuit est plus grande que 0 au moyen d'un `while`. Si ce n'est pas le cas, la fonction retourne `null`.

Dans le `while`, la valeur de `i` est `0`. Vous créez une variable `j` qui est égale à la longueur du nom (`visites`) et du signe `=`. La méthode `substring()` vérifie que le premier nom est celui qui est en argument de la fonction ; si c'est le cas, la position du premier caractère de la valeur est envoyée à la fonction `ingredients_biscuit()`, sinon `i` prend la valeur du mot qui suit immédiatement l'espace, c'est-à-dire le nom du deuxième cookie, et ainsi de suite jusqu'à ce que le test soit vrai.

La méthode substring()

La méthode `substring()` utilisée abondamment pour analyser la propriété `cookie` de l'objet `document` est nourrie par deux arguments : le premier donne la position du premier caractère et le second donne la longueur totale de la chaîne.

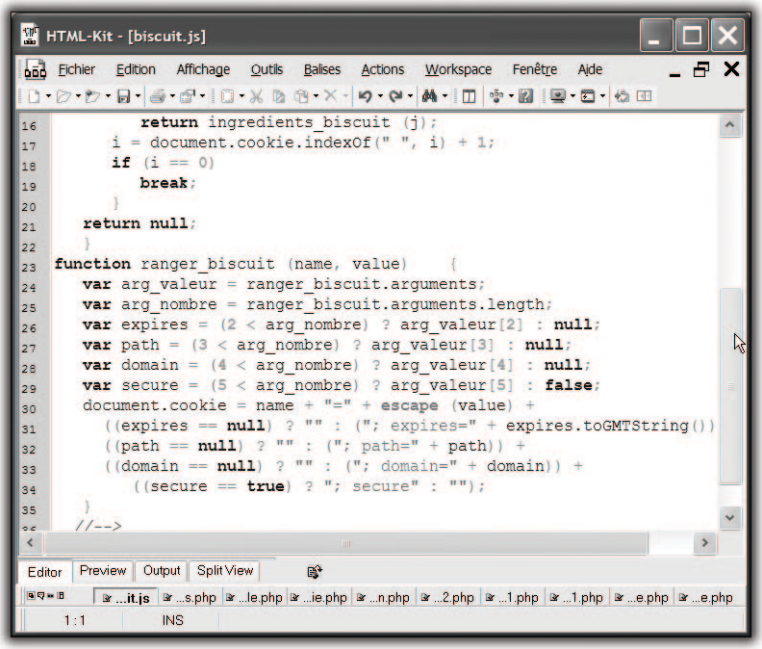
```
function manger_biscuit (nom) {
  var arg = nom + "=";
  var longueur_nom = arg.length;
  var taille_biscuit = document.cookie.length;
  var i = 0;
  while (i < taille_biscuit) {
    var j = i + longueur_nom;
    if (document.cookie.substring(i, j) == arg){
      return ingredients_biscuit (j);
    }
    else {
      i = document.cookie.indexOf(" ", i) + 1;
    }
    if (i == 0) {
     break;
    }
  } //fin du while
  return null;
}
```

Examinez maintenant la fonction `manger_biscuit()`.

```
function ingredients_biscuit (debut_mot) {
var fin_mot =
   document.cookie.indexOf (";", debut_mot);
if (fin_mot == -1) fin_mot = document.cookie.length;
  return unescape
```

```
   (document.cookie.substring(debut_mot, fin_mot));
}
```

Cette fonction reçoit le début du mot représentant la valeur et en localisant le point-virgule en induit la longueur. Elle renvoie la valeur reconstituée avec la méthode `unescape()`.



▲ Figure 9.24 : *Fonction ranger_biscuit()*

La dernière fonction a pour objectif de créer ou de modifier le cookie.

```
function ranger_biscuit (nom, valeur) {
var arg_valeur = ranger_biscuit.arguments;
var arg_nombre = ranger_biscuit.arguments.length;
var expires=(2<arg_nombre) ? arg_valeur[2]: null;
var path = (3 < arg_nombre) ? arg_valeur[3] : null;
var domain=(4<arg_nombre) ? arg_valeur[4]: null;
var secure=(5 < arg_nombre) ? arg_valeur[5]: false;
document.cookie = nom + "=" + escape (valeur) +
   ((expires == null) ? "" : ("; expires=" +
```

```
      expires.toGMTString())) +
   ((path == null) ? "" : ("; path=" + path)) +
   ((domain == null) ? "" : ("; domain=" + domain)) +
   ((secure == true) ? "; secure" : "");
}
```

Avec la détection du navigateur

En outre, JavaScript permet de détecter un navigateur.

```
var nom = navigator.appName;
var version = parseInt(navigator.appVersion);
var IDnavigateur = navigator.userAgent;
var navigateur = 0;
if (nom.indexOf("Explorer")>0){
  if (version >= 4) navigateur= 1; else navigateur=2;
  if (IDnavigateur.indexOf("MSIE 5.0")>0)navigateur=2;
}
if (navigateur ==0){
  var balise ='<link href="generique.css"
  rel="stylesheet" type="text/css" />\n";
}
else {
  var balise ='<link href="generique_IE.css"
  rel="stylesheet" type="text/css" />\n";
}
document.write (balise);
```

Le programme teste le nom et la version du navigateur et récrit la balise qui conduit à la feuille de style adéquate.

Le modèle objet de document (DOM) et AJAX

Le DOM (*Document Object Model*) est un arbre avec une hiérarchie descendante à partir de l'objet `document`. Il est matérialisé par les différentes balises du document HTML, XHTML et XML. La structure intériorisée dans le modèle est constituée de balises qui se côtoient et qui contiennent parfois d'autres balises. Ainsi tout le document est contenu dans les balises `<html>` et `</html>`, qui contiennent les balises `head` qui contiennent les balises `title` et les balises `<meta http-equiv= "Content-Type" content="text/html; charset=iso-8859-1"/>` et `<link rel="stylesheet" href="feuille.css" type="text/css" />`, tandis que la balise `body` est la balise frère de la balise `head` puisque ces deux balises sont au même niveau dans la hiérarchie.

Pour bien comprendre, il faut bien avoir l'arbre complet dans la tête et le visualiser. C'est un arbre généalogique : ainsi `head` est un descendant de `html`, un frère de `body`, un père de `link` et de `meta`.

Les éléments sont appelés des nœuds (`nodes`) en référence au langage XML qui est à l'origine de ce modèle. Les nœuds au-dessus du nœud courant sont appelés les ancêtres et les nœuds au-dessous sont les descendants. Chaque nœud se positionne dans l'arbre relativement aux autres nœuds.

AJAX

AJAX est vraiment au carrefour des langages du Web. Il utilise des passerelles entre le XML, le JavaScript, les CSS et d'autres langages comme PHP, Java, Ruby, etc. Des applications existent pour transformer des objets d'un langage à l'autre afin de les fournir à l'API JavaScript du navigateur qui pourra les manipuler. L'intérêt du JavaScript est qu'il agit directement sur la page sans demander d'autres manipulations et les changements se font à vue. Le DOM permet d'envoyer les données retournées, dans un élément de l'arbre HTML, c'est-à-dire à un endroit précis dans la page. Ces techniques évitent de solliciter le serveur à chaque requête et de faire travailler le navigateur sur le PC client sur des données serveur chargées en même temps que la page.

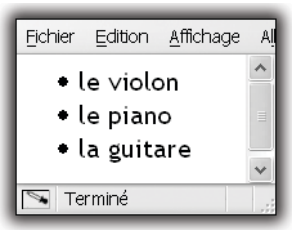
Au départ, JavaScript tournait sur lui-même : il était impossible de charger des données non incluses dans la page. Tout est parti de la classe JavaScript `XmlHttpRequest`, créée par Microsoft, à partir de la version 4.0 d'Internet Explorer, puis adoptée par les autres navigateurs.

Le principe d'AJAX (*Asynchronous JavaScript And XML*) est de pouvoir envoyer des requêtes et de les afficher sans recharger la page. La page devient alors une application. AJAX est une combinaison de technologies : HTML et CSS pour la présentation, DOM, qui transforme chaque élément de l'arbre en objet, et JavaScript avec l'objet `XMLHttpRequest` pour gérer les requêtes et les réponses. Asynchrone signifie que l'application, une fois la requête envoyée, n'attend pas la réponse mais poursuit son processus et affiche la réponse quand elle se présente.

Le nom de chaque propriété est écrit à la manière JavaScript. Ainsi `font-size` devient `fontSize`, et `background-color`, `backgroundColor`, etc.

Voici une liste d'instruments de musique :

```
<ul>
<li>le violon</li>
<li>le piano</li>
<li>la guitare</li>
</ul>
```

L'élément parent est la balise `ul` dont les enfants sont les balises `li` et les petits-enfants les nœuds texte de `li`, autrement dit leur contenu (le violon, le piano, la guitare).

◄ Figure 9.25 : *Liste non ordonnée*

Ce modèle DOM permet d'identifier les nœuds ou balises et donc d'agir dessus individuellement ou collectivement en adressant une commande à tous les nœuds de même niveau ou à tous les descendants d'une balise. Ces commandes peuvent influer sur le contenu d'un nœud ou sa position d'une manière dynamique. Avec les CSS, il est possible de masquer un nœud ou de le transformer, ou encore de le faire apparaître par un événement comme `:hover`.

◄ Figure 9.26 : *Titre noir*

```
  <h1 id="titre" class="titraille">Grand titre</h1>
  <button onclick="<button onclick=
"document.getElementById('titre')
.style.color = 'orange';">
Mettre le titre en orange</button>">Mettre le titre en orange
</button>
 </body>
</html>
```

◄ Figure 9.27 : *Titre orange*

L'élément peut être désigné par son identifiant (`getElementById`), par la valeur de son attribut `name` (`getElementByName`) ou par le nom de sa balise (`getElementByTagname`). Ici, le fait d'appuyer sur le bouton colorie le titre en orange sans que la page n'ait besoin d'être rechargée.

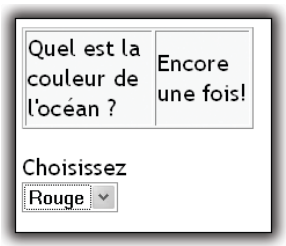
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD
HTML 4.01 Transitional//EN">
<html><head><title>Test</title>
<script type="text/javascript">
<!--
function contraste() {
 document.getElementsByTagName("body")[0]
.setAttribute("bgColor", "white");
 document.getElementsByTagName("body")[0]
.setAttribute("text", "black");
}
//-->
</script></head>
<body bgcolor="#FFFFCC" text="#E00000">
<button onClick="contraste()">En noir et blanc</button>
</pre></form>
</body></html>
```

◄ Figure 9.28 :
Noir et blanc

Ce bouton permet de rendre un site plus lisible en ce sens qu'il affecte un fond blanc et des caractères noirs à la page. Le premier élément `body` (`[0]`) est visé. La méthode `setAttribute()` donne une valeur à un attribut ou une propriété.

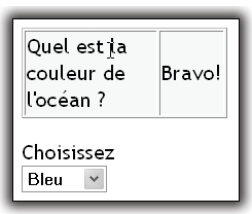
```
<html>
<head>
<script>
  function changeStyle()  {
    c = document.getElementById("cellule");
    c.style.width = "250px";
  }
</script>
<table border="1"><tr>
  <td id="cellule">C'est la cellule</td></tr>
</table>
<form>
  <input value="élargir"
    type="button"
    onclick="changeStyle();" />
</form>
</html>
```

Ici on insère une propriété `width` avec `getElementById()` et la fonction comprend deux lignes de code. Il est même possible de remplacer du texte dans la page, toujours sans la recharger. Dans le petit programme suivant, une question est posée et la réponse donnée via le menu déroulant du formulaire déclenche, par l'événement `onChange`, l'exécution de la fonction. Selon la réponse, la page affiche "*Encore une fois !*" ou "*Bravo !*" et le changement se fait à vue.



◄ Figure 9.29 :
Mauvaise réponse

```
<html><head><title>Test</title>
<script type="text/javascript">
<!--
function reponse() {
ind_choix=document.forms[0].choix.selectedIndex;
valeur_quatre=document.forms[0]
.choix.options[ind_choix].value;
if (valeur_quatre=='4'){
 var noeud_texte = document.createTextNode("Bravo!");
 }
 else {
var noeud_texte = document.createTextNode("Encore une fois!");
 }
 document.getElementById("lareponse")
.replaceChild(noeud_texte, document
.getElementById("lareponse").firstChild);
}
//-->
</script></head>
<body>
<table border="1" bgcolor="#FFFFC0" width="30%">
<tr><td id="z1">Quelle est la couleur de l'océan ?</td>
<td id="lareponse">ça dépend!</td></tr>
</table>
<p>
<form action="#" method="post">
Choisissez <br />
   <select name="choix" size="1" onChange="reponse()">
   <option value=""></option>
   <option value="1">Blanc</option>
   <option value="2">Noir</option>
   <option value="3">Rouge</option>
   <option value="4">Bleu</option>
   <option value="5">Jaune</option>
</select>
</form>

</body></html>
```



◄ Figure 9.30 : *Bonne réponse*

Toutes ces applications sont du DHTML utilisant le JavaScript et le DOM et parfois les CSS comme dans les menus dynamiques.

Un menu dynamique

▲ Figure 9.31 : *Listamatic*

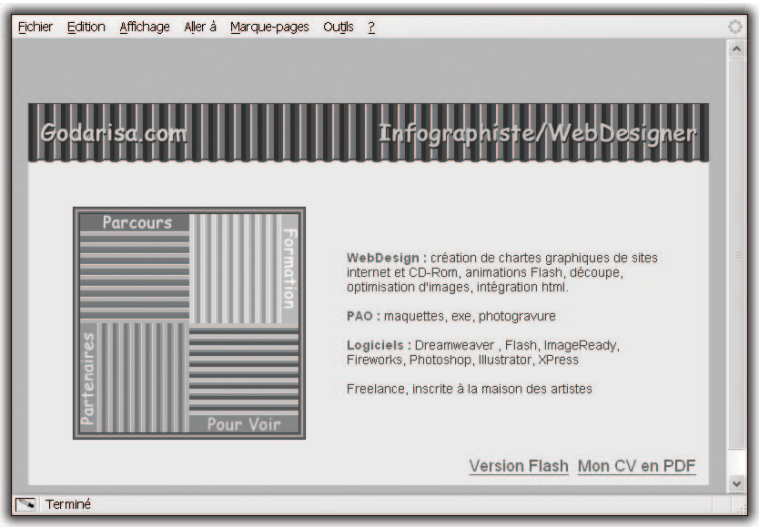
Vous pouvez réaliser facilement un menu en allant sur des sites comme `http://css.maxdesign.com.au/listamatic/`, qui présente toutes les façons de créer un menu dynamique, ou `www.accessify.com/tools-and-wizards/developer-tools/list-o-matic/`, qui génère le code d'un menu dynamique à partir d'un formulaire et de quelques choix de types de menus. Les possibilités vont du menu simple, avec juste un `:hover` au passage de la souris, au menu dynamique, qui s'enroule et se déroule, parfois avec l'aide du JavaScript (voir Figure 9.32).

Voici un menu en cascade que le site `www.ibilab.net` propose sur sa page `www.ibilab.net/webdev/exemples/menuencascade-exemple2.htm`. Ce menu a été adapté et implémenté sur le site biobearn.com par la graphiste Isabelle Godard (`www.godarisa.com`) (voir Figure 9.33).

▲ Figure 9.32 : *Accessify*

▲ Figure 9.33 : *Ibiliab*

Ce menu utilise du JavaScript et la propriété `display` avec les valeurs `none` et `block` pour afficher ou masquer le contenu des menus. Ces derniers sont placés horizontalement et déroulés verticalement.



▲ Figure 9.34 : *Godarisa*

Dans ces menus, on utilise le DHTML, le HTML dynamique composé de CSS, de JavaScript, de DOM. Les méthodes JavaScript `getElementsByTagName()` et `getElementById()` agissent directement sur des éléments de l'arbre DOM, la première par le biais du nom de la balise et la seconde par le biais de l'identifiant de l'élément.

```
<!-- modèle de menu sur : http://www.ibilab.net -->
<script type="text/javascript">
<!--
function hover(obj){
  if(document.all){
    UL = obj.getElementsByTagName('ul');
    if(UL.length > 0){
      sousMenu = UL[0].style;
    if(sousMenu.display == 'none' || sousMenu.display == ''){
       sousMenu.display = 'block';
     }else{
        sousMenu.display = 'none';
```

```
      }
    }
  }
}
function setHover(){
  LI = document.getElementById('menu')
.getElementsByTagName('li');
  nLI = LI.length;
  for(i=0; i < nLI; i++){
    LI[i].onmouseover = function(){
      hover(this);
    }
    LI[i].onmouseout = function(){
      hover(this);
    }
  }
}

function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a
.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}

function MM_preloadImages() { //v3.0
 var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments;
for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image;
d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.01
  var p,i,x;  if(!d) d=document;
  ✂ if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document;
n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n];
for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++)
  ✂ x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
var i,j=0,x,a=MM_swapImage.arguments;document.MM_sr=new Array;
 for(i=0;i<(a.length-2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x;
if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
//-->
```

```
</script>
<link href="menu.css" rel="stylesheet" type="text/css">
<link href="biobearn.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--
body {
 background-image:  url("images/fond.gif");
 background-repeat: repeat;
}
a {
 font-size: small;
 color: #339999;
 font-family: Arial, Helvetica, sans-serif;
}
a:visited {
 color: #339999;
}
a:hover {
 color: #cc6699;
}
a:active {
 color: #cc6699;
}
-->
</style>
</head>

<body onLoad="setHover();
MM_preloadImages('images/menu/menu1_on.gif'
,'images/menu/menu2_on.gif','images/menu/menu3_on.gif'
,'images/menu/menu4_on.gif')">
<table width="100%"  border="0" cellpadding="0"
cellspacing="0">
  <tr>
    <td align="center" valign="middle"><table width="760"
border="0" cellpadding="0" cellspacing="0" class="fondmenu">
      <tr>
        <td align="left" valign="top">
          <ul id="menu">
  <!-- logo -->
    <li class="first">
    <a href="index.php">
<img src="images/menu/logo_page.gif"
 width="140" height="35" hspace="0"
vspace="0" border="0" class="first"></a>
  </li>
    <!-- salade -->
  <li class="two">
    <img src="images/menu/menu1_salade.gif" width="40"
```

```
height="35" hspace="0" vspace="0" border="0">
  </li>

  <!-- debut du premier menu -->
  <li class="three">
<a href="#" onMouseOut="MM_swapImgRestore()"
onMouseOver="MM_swapImage('bouton1',''
,'images/menu/menu1_on.gif',1)">
<img src="images/menu/menu1_off.gif" name="bouton1"
width="89" height="35" hspace="0" vspace="0" border="0"></a>
    <ul onMouseOver="MM_swapImage('bouton1',''
,'images/menu/menu1_on.gif',1)"
onMouseOut="MM_swapImgRestore()">
<li><a href="implantation.php"><span>Implantations</span>
</a></li>
<li><a href="presentation.php"><span>Pr&eacute;sentation
</span></a></li>
<li><a href="promo.php"><span>Promos</span></a></li>
      <li><a href="service.php"><span>Services</span>
</a></li>
    </ul>
  </li>
<!-- fin du premier menu -->
```

La feuille de style joue sur le pseudo-élément `:hover` qui est relayé par l'événement JavaScript `onMouseOver` à cause des incompatibilités d'Internet Explorer. Certains propriétés sont destinées à Internet Explorer, comme `_display:inline` (notez le blanc souligné initial).

▲ Figure 9.35 : *Bio Béarn*

```
ul#menu li ul{
  display:none;
}

ul#menu li:hover ul{
  display:block;
}

ul#menu{
  margin:0px;
  padding:0px;
  list-style:none;
  float:left;
  font:small Arial;
}

ul#menu li{
  float:left;
  display:block;
}

ul#menu li.first{
  width: 140px;
}
ul#menu li.two{
  width: 40px;
}

ul#menu li.three{
  width: 89px;
}
ul#menu li.for{
  width: 32px;
}
ul#menu li.fiveseven{
  width: 99px;
}
ul#menu li.six{
  width: 35px;
}
ul#menu li.heigt{
  width: 59px;
}
ul#menu li.nine{
  width: 84px;
}
ul#menu li.ten{
  width: 83px;
}
```

```
ul#menu li a{
  display:block;
  color:#339999;
  text-decoration:none
}

ul#menu li a:hover{
  color:#CC6699;
}

ul#menu ul{
  position:absolute;
  background:#CCFFCC;
  list-style:none;
  margin:0px;
  padding:0px;
  width:140px;
  z-index:1;
}

ul#menu ul li{
  position:relative;
  float:none;
  margin:0px;
  padding:0px;
  _display:inline;
}

ul#menu ul li a{
  display:block;
  margin:0px;
  padding:0px;
  border:none;
}
ul#menu ul li a span{
  display:block;
  padding:1px 1px;
  border:none;
  cursor:hand;
}

ul#menu ul li a:hover span{
 color:#CC6699;
  border:none;
}

.fondmenu {
 background-color: #ccffcc;
```

```
 background-image:  url("images/menu/menu_fond.gif");
 background-repeat: no-repeat;
}
```

Un jour peut-être, il sera possible de créer ce genre de menu déroulant, au-dessus de la page, uniquement à l'aide des CSS, sans aucun apport d'autre langage.

9.2 Le PHP

Le PHP est principalement un langage de script qui est mélangé au code HTML pour exécuter des commandes avant que le fichier HTML soit présenté au navigateur. Le code PHP est exécuté par le serveur avant que le navigateur ne présente la page à l'internaute. Comme le résultat de tout cela est que tout le contenu de la page est du HTML, n'importe quel navigateur peut l'interpréter.

D'autres langages de script existent, comme ASP, JSP mais PHP est un langage open source simple d'approche, avec des développements qui permettent une grande souplesse dans des situations inédites. À l'instar de Java, il intègre la programmation objet et, comme il est beaucoup diffusé, de nombreux outils PHP particulièrement bien adaptés pour le Web sont développés. Comme il est gratuit, quasiment tous les hébergeurs web le propose.

Le PHP est fondé sur les phrases, à la manière d'un langage naturel. Le type par défaut est `string` (voir Figure 9.36).

Sans hébergeur, il est possible d'installer l'application EasyPHP. La procédure est simple. Téléchargez le logiciel sur `www.easyphp.org` puis un clic suffit pour installer le serveur Apache, le moteur PHP le plus récent et la base de données MySQL accompagnée de l'outil d'administration PHPmyAdmin. Sous Windows, vous avez déjà un serveur web installé : IIS. Si vous rencontrez un problème pour afficher la page d'accueil d'EasyPHP, c'est souvent à cause d'un conflit avec IIS, le serveur de Windows. Il suffit alors de le désactiver. De même, des logiciels pare-feu comme ZoneAlarm, par ailleurs utiles, peuvent empêcher l'accès à la page du logiciel d'administration de la base MySQL, PHPmyAdmin. Dans la plupart des cas, EasyPHP ne pose aucun problème.

▲ Figure 9.36 : *EasyPHP*

Ma page

Ouvrez votre traitement de texte ou d'édition de code. Entrez les balises obligatoires `<html></html>`. À l'intérieur, ajoutez les balises qui composent les deux parties d'une page : l'en-tête `<head></head>` et, au-dessous, les balises `<body></body>`. Vous obtenez une page blanche. Maintenant, ajoutez un titre dans les balises `<head></head>`, qui apparaîtra comme une enseigne dans l'en-tête du navigateur : `<title>Le club</title>`.

La page est encore vide. Vous allez vous présenter. Entre les balises `<body></body>`, écrivez `je m'appelle`.

Saisissez maintenant votre premier code PHP. Le code PHP doit être entouré des balises `<? ?>` (ou parfois `<?php ?>` ou `<% %>`). Avant la balise `<html>`, insérez votre première variable.

```
<?
$mon_nom="François";
?>
```

Le caractère $ devant un mot indique qu'il s'agit d'une variable. Une variable est un espace de mémoire vive alloué à une information. Donnez-lui de préférence un nom compréhensible. `mon_nom` signifie davantage qu'une adresse mémoire avec du code informatique. Quand vous relirez votre programme, vous saurez qu'il s'agit du nom de quelqu'un. Ici, l'espace souligné remplace l'espace, qui ne doit pas figurer dans un nom de variable.

Le caractère "égal" crée une affectation. Les guillemets indiquent au moteur PHP qu'il s'agit d'une chaîne de caractères et le point-virgule ferme la commande. La commande peut s'étaler sur plusieurs lignes. Le point-virgule indique au moteur que ce qui a précédé est une unité, un ordre d'exécution. L'affectation consiste à affecter une valeur à une variable. Ainsi, l'espace mémoire appelé `mon_nom` reçoit la valeur `François`.

Le caractère *éga*"

Le fait que les premiers concepteurs de programmes aient choisi le caractère *égal* (=) pour caractériser l'affectation d'une valeur à une variable entraîne souvent la confusion avec le test d'égalité qui, lui, est signalé par un *double égal* (==). Certains bogues sont dus à cette confusion, surtout dans les tests avec `if`.

Enregistrez maintenant votre page dans le répertoire *club* à l'intérieur du répertoire *www* du dossier d'EasyPHP sous le nom *index.php*. Puis ouvrez votre navigateur web et cliquez du bouton droit sur la petite icône d'EasyPHP dans la barre de tâches de Windows. Un menu apparaît. Sélectionnez *Web local* et cliquez sur le répertoire *club*. Rien n'apparaît, juste une page blanche. Vous avez créé une variable mais ne l'avez pas affichée sur l'écran. Pour cela, inscrivez `<? echo $mon_nom ?>` en dessous de la balise `body`. Remarquez l'omission du point-virgule. En fait, la balise `?>` signale la fin de la commande et le point-virgule est ici optionnel.

Maintenant, vous avez plusieurs méthodes pour ajouter le nom :

- Première solution :

```
$mon_nom="François Rabelais";
```

- Deuxième solution :

```
$mon_nom="François"." "."Rabelais";
```

- Troisième solution :

```
$mon_nom = "François";
$mon_nom =$mon_nom." Rabelais";
```

- Quatrième solution :

```
$mon_nom = "François";
$mon_nom="$mon_nom Rabelais";
```

- Cinquième solution :

```
$mon_nom = "François ";
$mon_nom.="Rabelais";
```

Ces cinq solutions donnent le même résultat : `François Rabelais`.

◄ Figure 9.37 : *François Rabelais*

Passons maintenant aux explications.

`$mon_nom="François Rabelais";` est une chaîne de caractères. Vous écrivez comme d'habitude, en langage naturel, avec un espace entre les mots.

`$mon_nom="François"." "."Rabelais";` introduit la fonction de concaténation, exprimée par le point. Cette fonction assemble trois tronçons, espace inclus. Les guillemets, comme les apostrophes, indiquent que la valeur est de type `string`. Dans certains langages, vous devez toujours préciser le type d'une donnée et il est malaisé de faire passer une donnée d'un type à un autre, ce qui n'est pas le cas en langage PHP. Il reconnaît de lui-même le type de la donnée.

Il existe des différences entre une paire de guillemets et une paire d'apostrophes. Par exemple, si vous écrivez `$mon_nom='$mon_nom Rabelais'` au lieu de `$mon_nom="$mon_nom Rabelais"`, vous affichez littéralement `$mon_nom Rabelais`. La variable n'a pas été extrapolée. Elle a été prise pour un mot commençant par le caractère "dollar".

Les variables scalaires

Les variables scalaires contiennent une seule valeur, par opposition aux tableaux qui en contiennent plusieurs.

Le nom des variables

Un nom de variable qui représente une valeur ou un ensemble de valeurs doit commencer par une lettre ou un blanc souligné (_), suivi de lettres, de chiffres ou de blancs soulignés. Attention, le langage PHP distingue la casse dans les nom de variable. Ainsi `$Fromage` et `$fromage` sont des variables différentes.

Choix du nom des variables

Soyez cohérent dans le choix du nom des variables. L'objectif est d'obtenir un nom évocateur, mais pas trop long. `$nbre_bouteille` est aussi significatif que `$nombre_de_bouteilles` mais vous aurez moins de caractères à saisir à chaque fois que vous l'utiliserez. Soyez rigoureux dans vos critères de nommage. Évitez les capitales car vous risquez d'oublier que vous les avez employées et d'engendrer ainsi une erreur avec une variable inexistante. Notez que le nom d'un champ de formulaire sert aussi à nommer la variable qui en est issue ; soyez rigoureux aussi sur le choix de ces noms de champ.

La valeur des variables scalaires peut être de plusieurs types. Il est bon de connaître ces derniers même si, en général, le langage PHP fait le travail à votre place dans la plus grande transparence.

Le type des variables

Tab. 9.3 : Les différents types de variables

Type	En français	Exemple
`integer` ou `int`	Entier	`45`
`float`	Nombre à virgule	`45.2`
`string`	Chaîne de caractères	`"J'veux faire du PHP!"`
`boolean` ou `bool`	Booléen	`false` ou `true`

Les variables en langage PHP sont typées implicitement. Ainsi, PHP reconnaît un chiffre ou des lettres et le point du nombre à virgule flottante. Il vous laisse cependant la possibilité de forcer le type :

```
$code_postal="06400";
$code_postal= (int)$code_postal;
```

Ici, la variable `$code_postal`, qui serait naturellement une chaîne de caractères à cause des guillemets, peut être transtypée en nombre. Le forçage du type en `int` lui enlève le 0 du début. Par contre, si 06400 avait été sans guillemets, le nombre aurait interprété comme un octal à cause du 0 initial et aurait donc valu 3328 à la deuxième ligne. Avec l'associativité des opérateurs, le sens du traitement de l'affectation et du transtypage est de la droite vers la gauche. La variable `$code_postal` est donc ici transtypée en `integer` puis affectée à elle-même : elle n'a plus rien d'un code postal. Faites bien la différence entre un chiffre et un nombre, sinon, vous aurez des surprises.

Les entiers

Les variables de type entier sont assez simples à comprendre.

Vous pouvez écrire par exemple :

```
$a=1;
$b=2;
$c=$a+$b;
```

En l'occurrence, $c contient la valeur 3.

Si vous mettez les valeurs 1 et 2 entre guillemets ou apostrophes, PHP devine qu'il s'agit de valeurs numériques, et non de chaînes de caractères, grâce à l'opérateur +. Il fera la somme des deux variables :

```
$a='1'; $b="2"; $c=$a+$b;
```

PHP vous simplifie la vie en assumant ces tâches de typage des données qui alourdissent inutilement le travail du programmeur et peuvent constituer une source d'erreurs.

Les nombres à virgule flottante

Méfiez-vous des float, c'est-à-dire des nombres à virgule flottante, car ils sont sujets aux approximations dans les résultats. Si vous avez 6.555973333 comme résultat d'une opération, l'ordinateur arrêtera le résultat sur une approximation pour éviter de générer un bogue de programme sans fin.

En anglais, le point remplace la virgule utilisée en français.

Les booléens

Pour spécifier un booléen, vous pouvez utiliser false ou true (peu importe la casse).

Les valeurs considérées comme false sont :

- le booléen false ;
- l'entier 0 (zéro) ;
- le nombre à virgule flottante 0.0 (zéro) ;
- la chaîne de caractères vide et le caractère 0 ;
- un tableau sans élément ;
- le type null (y compris les variables passées par la fonction unset()).

Les autres valeurs sont considérées comme true.

Si la variable $variable a l'une des valeurs false précédentes, l'évaluation du test if(!$variable) est true. Par ailleurs, if ($varia-

ble) est true si la variable existe et qu'elle soit différente de 0, de null ou de "". Si elle vaut " ", l'expression renvoie aussi la valeur true.

Les chaînes de caractères

Le type string est de loin le plus utilisé. Les espaces sont significatifs à l'intérieur d'une chaîne de caractères. Elle est encadrée par des guillemets droits ou doubles (guillemets anglais).

Créer un aiguillage de feuilles de style

Une solution de rechange quand on ne trouve aucune autre solution est de créer une feuille de style pour chaque navigateur. C'est quand même une solution extrême car elle ne répond plus aux critères qui conduisent à l'emploi des CSS. La maintenance est plus malaisée car il faut alors modifier plusieurs feuilles de style et les valider, ce qui peut prendre du temps. Il existe quand même des situations où le développeur doit sortir une nouvelle version de son site rapidement. Il lui reste alors cette solution.

Il faut commencer par nommer toutes les pages avec l'extension *.php* et modifier les liens en conséquence.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
<html>
<head>
<title>Ecoute s'il pleut : Un séjour magique en toute
liberté! Accueil</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />
<?
if (eregi('msie', $HTTP_USER_AGENT) && !eregi('opera',
$HTTP_USER_AGENT))
{
// Internet Explorer
echo '<link href="ie.css" rel="stylesheet"
type="text/css" />';
}
elseif (eregi('opera', $HTTP_USER_AGENT))
{
// Opera
echo '<link href="opera.css" rel="stylesheet"
type="text/css" />';
```

```
}
elseif (eregi('Mozilla/4.', $HTTP_USER_AGENT))
{
// Netscape 4.x
echo '<link href="ns4.css" rel="stylesheet"
type="text/css" />';
}
elseif (eregi('Mozilla/5.0', $HTTP_USER_AGENT) &&
!eregi('Konqueror', $HTTP_USER_AGENT))
{
// Netscape 6
echo '<link href="ns6.css" rel="stylesheet"
type="text/css" />';
}
else
{
// Autres navigateurs
echo '<link href="defaut.css" rel="stylesheet"
type="text/css" />';
}
?>
</head>
```

L'aiguillage est créé avec des `if`. La variable d'environnement `$HTTP_USER_AGENT` est testée. Selon les caractères qu'elle contient, le lien vers la feuille de style est récrit dynamiquement.

Tout ce code sera caché dans la feuille de style, seul le résultat sera visible.

9.3 Le XML

Le XML est, comme le HTML, un descendant de l'ancêtre du langage Internet : le SGML. Ce dernier était très compliqué à mettre en œuvre. Il a donné naissance à des simplifications sous la forme du HTML puis du XML. Le HTML est devenu un langage de description de page très libre. Ainsi les balises HTML peuvent être mises en capitales ou en minuscules. La balise d'ouverture peut être en capitales et celle de fermeture en bas de casse. Par ailleurs, une balise d'ouverture peut ne pas comporter de balise de fermeture, comme la balise <p>. La concurrence entre les différents navigateurs a encore ajouté de la confusion, chacun créant ses propres balises. En outre, le Web s'est diversifié, se répandant sur les téléphones portables, les assistants numériques. Il a donc fallu trouver un langage plus rigoureux qui permettait, à partir d'un même contenu, de visionner des pages à l'aspect différent.

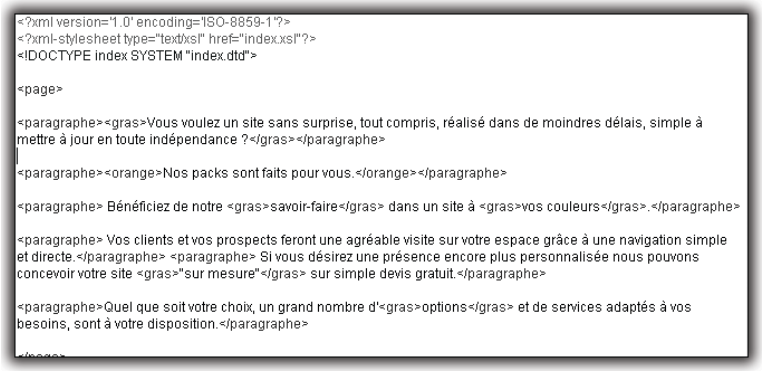
▲ Figure 9.38 : *Le site XMLfr*

Aux débuts d'Internet, la simplicité et le manque de contraintes du HTML ont contribué à le répandre très largement et rapidement à travers le monde et à en faire un standard. Quand les sites web ont grossi et sont devenus professionnels, le langage HTML a montré ses limites car, s'il caractérise la forme du document, il ne décrit pas le contenu de la page. Par ailleurs, il était difficile de mettre à jour les pages et de s'y retrouver dans un code mélangeant la forme et le contenu.

Qu'est-ce que le XML ?

C'est un langage dont les balises caractérisent le contenu. Il est rigoureux et n'accepte aucune faute de syntaxe. Les balises d'ouverture et de fermeture doivent avoir la même casse. Une balise solitaire doit être accompagnée d'une barre oblique (par exemple, `<br />`). Comme les

CSS, le XML sépare la présentation et les données. Un document (le fichier DTD) définit les balises et leur syntaxe, et facilite la construction et le débogage du fichier *.xml* conformément à cette DTD



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="index.xsl"?>
<!DOCTYPE index SYSTEM "index.dtd">

<page>

<paragraphe><gras>Vous voulez un site sans surprise, tout compris, réalisé dans de moindres délais, simple à
mettre à jour en toute indépendance ?</gras></paragraphe>

<paragraphe><orange>Nos packs sont faits pour vous.</orange></paragraphe>

<paragraphe> Bénéficiez de notre <gras>savoir-faire</gras> dans un site à <gras>vos couleurs</gras>.</paragraphe>

<paragraphe> Vos clients et vos prospects feront une agréable visite sur votre espace grâce à une navigation simple
et directe.</paragraphe> <paragraphe> Si vous désirez une présence encore plus personnalisée nous pouvons
concevoir votre site <gras>"sur mesure"</gras> sur simple devis gratuit.</paragraphe>

<paragraphe>Quel que soit votre choix, un grand nombre d'<gras>options</gras> et de services adaptés à vos
besoins, sont à votre disposition.</paragraphe>
```

▲ Figure 9.39 : *Fichier XML*

Un document XML appelle plusieurs documents :

- un fichier avec une extension *.xml*, qui stocke les données avec les balises caractérisant chaque type de données ;
- un fichier XSLT (Extensible Stylesheet Transformations), avec l'extension *.xsl*, qui est la feuille de style et qui décrit l'apparence de la page et la forme de chaque contenu ;
- un fichier *.dtd*, qui décrit la syntaxe, donne la structure du contenu et permet de déceler les erreurs ;
- un document qui appelle tous ces fichiers, en l'occurrence un document écrit dans un langage dynamique, comme PHP ou Java, ou même HTML avec du JavaScript inséré.

Dans le cas du XML, les navigateurs sont encore plus déterminants pour le code. C'est le principal écueil.

La différence entre le XSLT et les CSS

La principale différence entre le XSLT et les CSS est qu'un document XSLT contient du code qui agit directement sur les données et leur présentation, avec des modèles qui incluent des boucles et diverses

opérations, comme la concaténation. Les modèles sont des unités de programme et de présentation. Le modèle est d'abord appelé :

```
<xsl:template match="gras">
<b><xsl:value-of select="."/></b>
</xsl:template>
```

Cela se traduirait ainsi en CSS :

```
.gras {
font-weight: bold ;
}
```

Le fichier XSLT est un document XML qui intègre le code XHTML avec des appels de modèle :

```
<td valign="top">
<xsl:apply-templates select="page/paragraphe"/>
</td>
```

À n'importe quel endroit dans la feuille XSLT, le modèle appelé est défini :

```
<xsl:template match="paragraphe">
<p class="courant">
<xsl:apply-templates/>
</p>
</xsl:template>
```

La feuille de style qui l'accompagne définit les propriétés de `courant` :

```
.courant {
font-family: "Arial", "Helvetica", "sans-serif";
font-size: 14px;
color: #4087C1
}
```

La feuille XSLT est plus intéressante que la feuille de style car elle offre plus de possibilités, même si on peut raisonnablement penser qu'un jour elles se rejoindront (voir Figure 9.40).

La feuille XSLT donne plus de latitude car elle agit sur la présentation des données, mais aussi sur leur répétition ou leur déclinaison avec les

commandes XPath. Le XSLT possède aussi une recherche de motifs plus avancée. page/paragraphe cherche paragraphe qui est enfant de page.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html" encoding="ISO-8859-1"/>


<xsl:template match="/">
<html lang="fr" xml:lang="fr" xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>AcceSites</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>

<script language="JavaScript"><xsl:comment><![CDATA[

function MM_swapImgRestore() { //v3.0
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}

function MM_preloadImages() { //v3.0
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}

function MM_findObj(n, d) { //v4.0
  var p,i,x;  if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
    d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && document.getElementById) x=document.getElementById(n); return x;
}

function MM_swapImage() { //v3.0
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
   if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}

]]></xsl:comment>
</script>

<link rel="stylesheet" href="accessites.css" type="text/css"/>
```

▲ Figure 9.40 : *Fichier XSLT*

Le fichier XML

Le fichier XML est composé de deux parties complémentaires.

L'espace de nommage

Commencez par déclarer la version XML utilisée pour stocker les données ainsi que le type d'encodage.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

Ce type d'encodage permet de rendre correctement les caractères propres à l'Europe occidentale.

À la ligne suivante, déclarez la DTD.

```
<!DOCTYPE nomDTD SYSTEM "nom.dtd">
```

Nommez la DTD et indiquez le chemin vers le fichier ou l'URL. Ensuite déclarez le fichier XSLT qui mettra le fichier XML en forme.

```
<?xml-stylesheet type="text/xsl" href="adherents.xsl"?>
```

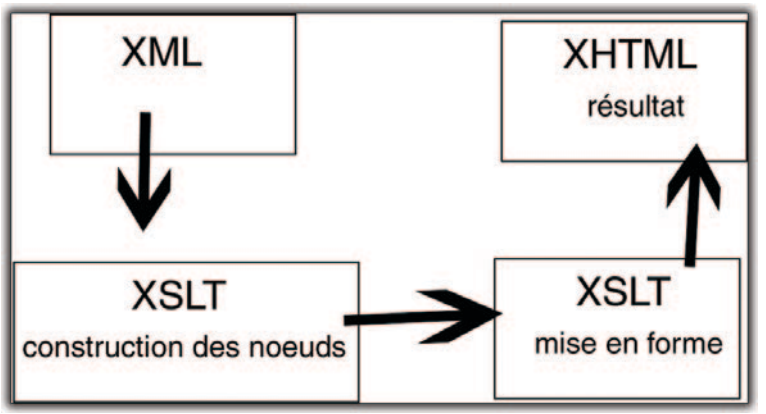
Les éléments et les attributs

Le fichier XML est formé de balises comme dans l'exemple suivant :

```
<fiche>
<img src="jeandupond.jpg"/>
<prênom>Jean<prênom>
<nom>Dupont</nom>
<adresse>16, rue du Pont Louis-Philippe</adresse>
<codepostal>75004</codepostal>
<ville>Paris</ville>
</fiche>
```

Chaque balise ouvrante renvoie à une balise fermante rigoureusement positionnée pour encadrer les données, excepté la balise `img` qui, ici, se termine par / du fait que c'est un élément isolé. Les balises sont nommées "éléments" tandis que leurs propriétés, comme la propriété `src` de l'élément `img`, sont appelées "attributs". L'élément contient du texte.

Le fichier XML doit avoir la bonne forme. Si vous choisissez des capitales pour nommer les balises, vous devrez vous y tenir, qu'elles soient ouvrantes ou fermantes. Vous allez donc créer un fichier de DTD dans lequel vous consignerez les règles que vous appliquerez à vos balises. Il servira à valider votre fichier et à repérer les erreurs de syntaxe. Vous pourriez ainsi décider que telle balise est obligatoirement à l'intérieur de telle autre balise.



▲ Figure 9.41 : *Processus XSLT*

Un fichier XML est un fichier brut de données, sans aucune présentation, mais ponctué de balises. Il est accompagné d'un fichier XSLT. Ce fichier de présentation est parcouru et interprété par une application serveur qui va fusionner les fichiers pour fabriquer une page qui sera soit un vrai fichier, soit l'écran par défaut. Le langage XSLT est déconcertant pour tout programmeur formé aux langages procéduraux ou orientés objet. Il est basé sur la création de règles de modèle (template) à partir du fichier source. Les transformations se font sur l'arbre résultat pour présenter les données d'une manière lisible par tous.

Le fichier XML est traité (parsed) par un analyseur XML qui est, par défaut, le module Expat. Ce fichier correspond au contenu de la page. C'est ce document qui sera modifié dans le cadre d'une mise à jour. Comme les balises sont explicites et le contenu écrit sans codage, n'importe quel utilisateur connaissant deux ou trois règles saura modifier ce type de fichier sans dommages.

Tab. 9.4 : Codage des caractères dans un fichier XML

Caractère	Codage	Statut
&	`&`	Obligatoire
<	`<`	Obligatoire
>	`>`	Obligatoire

Tab. 9.4 : Codage des caractères dans un fichier XML

Caractère	Codage	Statut
'	`'`	À l'intérieur d'un attribut délimité par des apostrophes
"	`"`	À l'intérieur d'un attribut délimité par des guillemets

Voici quelques règles :

- Les balises doivent être de la même casse, généralement en minuscules.
- Toute balise ouvrante doit être accompagnée par une balise fermante.
- Le nom d'une balise ou d'un attribut doit débuter par une lettre ou un blanc souligné (_), suivi éventuellement par plusieurs lettres, chiffres, points, tirets ou blancs soulignés.
- Une balise sans contenu se termine par />, par exemple `<br/>` ou `<br />`.
- Les attributs sont entre guillemets ou entre apostrophes, par exemple `<page numéro="1">`.
- Les balises ouvrantes et fermantes de différentes entités ne peuvent se chevaucher : elles doivent s'emboîter ou se suivre.
- Le caractère "guillemet" est codé quand il appartient à un attribut délimité par des guillemets.
- Le caractère "apostrophe" est codé quand il appartient à un attribut délimité par des apostrophes.
- Les noms de balise ou d'attribut ne peuvent contenir d'espace.
- Les balises ou les attributs ne peuvent avoir un nom qui commence par `xml` ou `XML` ou `Xml`, etc.

L'auteur d'un fichier XML nomme les différentes balises de sorte que le sens du contenu soit explicite pour un utilisateur.

```
<titre>Php &amp; MySQL en ligne
<sous-titre>L'essentiel pour concevoir
un site web dynamique!</sous-titre>
<collection référence="3679">e-poche</collection></titre>
```

Ce que nous venons d'énoncer est essentiel pour que le document XML soit dans une bonne forme. La DTD qui accompagne ce fichier permettra de le valider.

La DTD

La DTD est un document qui édicte les règles relatives aux entités composant le fichier XML, c'est-à-dire qu'elle décrit la succession et l'imbrication des éléments et des attributs.

Les différents types de déclarations

On déclare les différents éléments et attributs en décrivant leur contenu.

Tab. 9.5 : Différentes déclarations dans une DTD

Déclaration	Explication
`<!ELEMENT titre (#PCDATA)>`	L'élément `titre` contient du texte traité.
`<!ELEMENT titre (sous-titre, collection)>`	L'élément titre contient les éléments `sous-titre` et `collection`.
`<!ELEMENT titre (sous-titre, collection+)>`	L'élément `titre` contient l'élément `sous-titre` et une à plusieurs balises `collection`.
`<!ATTLIST collection référence CDATA>`	L'élément `collection` a comme attribut `référence` de type `CDATA` (texte).

Prenons par exemple un fichier XML tiré du site `www.accesites.com`.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<?xml-stylesheet type="text/xsl" href="index.xsl"?>
<!DOCTYPE index SYSTEM "index.dtd">

<page>

<paragraphe><gras>Vous voulez un site sans surprise,
tout compris, réalisé dans de moindres délais, simple à
mettre à jour en toute indépendance ?</gras></paragraphe>

<paragraphe><orange>Nos packs sont faits pour vous.</orange>
</paragraphe>
```

```
<paragraphe> Bénéficiez de notre <gras>savoir-faire</gras>
dans un site à <gras>vos couleurs</gras>.</paragraphe>

<paragraphe> Vos clients et vos prospects feront une agréable
visite sur votre espace grâce à une navigation simple et
directe.</paragraphe> <paragraphe> Si vous désirez une
présence encore plus personnalisée nous pouvons concevoir
votre site <gras>"sur mesure"</gras> sur simple devis gratuit.
</paragraphe>

<paragraphe>Quel que soit votre choix, un grand nombre
d'<gras>options</gras> et de services adaptés à vos besoins,
sont à votre disposition.</paragraphe>

</page>
```

Remarquez les balises imbriquées. Les données `paragraphe` contiennent parfois les balises `orange` ou `gras`, mais aussi du texte.

▲ Figure 9.42 : *Accesites.com*

Voici la DTD :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT page (paragraphe+)>
<!ELEMENT paragraphe (gras*,orange*)>
<!ELEMENT gras (#PCDATA)>
<!ELEMENT orange (#PCDATA)>
```

L'élément `page` contient un ou plusieurs éléments `paragraphe`. L'élément `paragraphe`, lui, contient aucun ou plusieurs éléments `gras` et `orange`. Les éléments `gras` et `orange` sont des données traitées par le processeur XSLT.

Le XSLT

Le XSLT est un langage de type XML, qui utilise la syntaxe et la sémantique d'un autre langage nommé XPath. Tout cela forme un vrai langage de programmation, qui permet de créer des processus de grande complexité.

L'espace de nommage

Il s'agit d'une sorte d'en-tête du fichier. Il contient le type d'encodage des caractères (ici les caractères d'Europe occidentale) et la version de XML (en ce moment, la 1.0).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
version="1.0">
```

La déclaration `xmlns` (XML name space) a un attribut qui ressemble à une URL, mais en fait, c'est juste une convention qui désigne le type de nommage qui sera utilisé.

La feuille de style

La feuille de style est contenue entre les balises `<xsl:stylesheet…>` et `</xsl:stylesheet>` dans laquelle vous pouvez déclarer les différents modèles en commençant par le modèle de la racine. Ce modèle représente la forme générale de la page. Ainsi, vous y mettrez votre charte graphique : le logo, la barre de menus, etc. tout ce qui enveloppe les

données. Vous partez d'un fichier HTML que vous normalisez en XHTML. Vous convertissez alors les balises <br> en <br /> selon les règles énoncées plus haut. Ensuite vous caractérisez le HTML comme étant :

```
<html lang="fr" xml:lang="fr" xmlns="http://www.w3.org/1999/xhtml">
```

Le fichier HTML original peut contenir des caractères que vous ne voulez pas soumettre au traitement XSLT, par exemple votre code JavaScript dans l'en-tête. Dans ce cas, il faut l'introduire dans la balise <![CDATA[…]]>.

```
<script language="JavaScript"><xsl:comment><![CDATA[
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){
if(!d.MM_p) d.MM_p=new Array();
  var i,j=d.MM_p.length,a=MM_preloadImages.arguments;
for(i=0; i<a.length; i++)
  if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++]
.src=a[i];}}
}
]]></xsl:comment>
</script>
```

Cela n'empêche pas d'utiliser aussi des feuilles de style dans le document résultat.

```
<link rel="stylesheet" href="accesites.css" type="text/css"/>
```

Les modèles (templates)

Votre modèle est encadré par les balises <xsl:template> et </xsl:template>. L'attribut match désigne l'élément qui correspond au modèle dans le fichier XML :

```
<xsl:template match="/">
```

Ici le caractère / désigne l'élément racine.

Vous introduisez votre code entre les balises. À l'endroit où vous devez introduire des données XML qui sont sous l'arborescence du modèle courant, vous appelez le modèle qui leur correspond (match) soit par son

nom de balise, soit par un terme générique. Voici un exemple expurgé de nombreuses lignes de code JavaScript et HTML car ici seul le code XSLT nous intéresse. Dans le fichier complet, l'appel au modèle de l'élément `paragraphe` est inséré dans un tableau HTML.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl
="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html" encoding="ISO-8859-1"/>
<xsl:template match="/">
<html lang="fr" xml:lang="fr"
xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>AcceSites</title>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"/>

<script language="JavaScript"><xsl:comment><![CDATA[
function MM_preloadImages() { //v3.0
  var d=document; if(d.images){
if(!d.MM_p) d.MM_p=new Array();
    var i,j=d.MM_p.length,a=MM_preloadImages.arguments;
for(i=0; i<a.length; i++)
    if (a[i].indexOf("#")!=0){ d.MM_p[j]
=new Image; d.MM_p[j++]
.src=a[i];}}
}
]]></xsl:comment>
</script>

<link rel="stylesheet" href="accessites.css" type="text/css"/>
</head>
<body bgcolor="#FFFFFF" text="#000000" leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0"
onLoad="MM_preloadImages('images/boutpack_over.gif',
'images/boutoption_over.gif','images/boutcommand_over.gif',
'images/boutcontact_over.gif','images/boutmarche_on.gif',
'images/boutdevis_on.gif')">

 <xsl:apply-templates select="page/paragraphe"/>

</body>
</html>
</xsl:template>
```

Ensuite nous ajoutons les modèles des différents éléments :

```
<xsl:template match="paragraphe">
  <p class="courant">
<xsl:apply-templates/>
</p>
  </xsl:template>
```

Ici il s'agit d'un paragraphe. Nous utilisons donc les balises et . <xsl:apply-templates/> est une commande générique qui évite de nommer chaque modèle à appeler et qui appelle donc le contenu du paragraphe, dont les contenus gras et orange, en vérifiant leur modèle.

```
<xsl:template match="gras">
 <b><xsl:value-of select="."/></b>
  </xsl:template>
  <xsl:template match="orange">
 <p class="sstitre"><xsl:value-of select="."/></p>
  </xsl:template>
```

La commande <xsl:value-of select="."/> représente la valeur de l'élément courant. On aurait pu aussi nommer chaque élément.

Comme vous pouvez le constater, nous pouvons envisager la structure de données d'un fichier XML comme une arborescence ou une imbrication de données de type "poupées russes". C'est un langage déclaratif fondé sur des modèles.

Dans l'exemple, nous voyons que dans un document XSLT qui obéit au format XML, qui est strict, il est toutefois possible d'intégrer du JavaScript, qui lui n'obéit pas aux règles du XSLT. Il est alors donné en commentaire.

Notre ouvrage n'est pas un livre sur le langage XML. Il s'agit donc là d'une introduction pour vous aider à aborder ce langage à travers des exemples pratiques.

Le XSLT et les CSS

Le fichier XSLT est un fichier XHTML avec, en plus, une DTD différente et des commandes XPath. Il est donc possible d'y glisser un lien vers une feuille de style.

La feuille de style en cascade est assez simple.

```
.courant {
font-family: "Arial", "Helvetica", "sans-serif";
font-size: 14px;
color: #4087C1
}
.sstitre {
font-family: "Arial", "Helvetica", "sans-serif";
font-size: 16px;
color: #FF9999;
font-weight: bold
}
.titre {
font-family: "Arial", "Helvetica", "sans-serif";
font-size: 24px;
color: #4087C1;
font-weight: bold
}
.titre_orange {
font-family: "Arial", "Helvetica", "sans-serif";
font-size: 24px;
color: #FF9999;
font-weight: bold
}
```

Elle porte surtout sur les caractéristiques de certains caractères. Il est possible de créer les balises à volonté, à condition de les inscrire dans la DTD.

Le fichier PHP

Le fichier PHP appelle les différents fichiers qu'il envoie vers les API pour qu'ils soient traités en vue d'un fichier résultat. Voici un exemple de code :

```
$xslt=xslt_create();
$resultat=xslt_process($xslt, 'index.xml', 'index.xsl');
echo $resultat;
xslt_free($xslt);
```

Le programme crée un analyseur XSLT par la fonction `xslt_create()`, puis il crée, avec l'analyseur, le fichier XML et le fichier XSLT, un fichier résultat qui sera affiché ici à l'écran avant d'être effacé de la mémoire du serveur. Cette fonction a trois arguments supplémentaires optionnels, qui permettent de passer des paramètres au processeur (ici Sablotron) :

- Le premier argument est le nom du fichier résultat, si vous voulez conserver ce résultat dans un fichier. Si vous ne l'utilisez pas mais que vous utilisiez les arguments suivants, il faut mettre `null` pour signaler son absence.
- Le deuxième argument est un tableau d'arguments qui, pour l'instant, est réservé aux versions prochaines de Sablotron.
- Le troisième est un tableau de paramètres pour le processeur XSLT, comme des paramètres de pagination.

Avant de lancer l'analyseur XSLT, le fichier PHP peut procéder à des opérations. Tous ces fichiers se marient harmonieusement en se complétant. PHP apporte sa richesse de fonctions et sa puissance, XML structure les données et les rend simples à mettre à jour, y compris par des non-spécialistes, XSLT ajoute une grande diversité de formats du même document (fichier HTML, fichier CSV…).

9.4 Conclusion

Le Web est vraiment le carrefour de toutes les technologies. De plus en plus, l'architecture du Web est utilisée pour les applications à l'intérieur des entreprises. Dans cette architecture, les CSS resteront un élément important par leur simplicité de conception. Les navigateurs se normaliseront, sous peine de devenir marginaux, et les CSS donneront alors toute leur puissance et déploieront toutes leurs potentialités.

10

Références

Voici quelques sites autour des technologies évoquées dans ce livre.

10.1 CSS

Cette liste n'est pas exhaustive. Cependant, vous rendrez régulièrement visite aux sites qui suivent si vous vous intéressez aux CSS.

Yoyodesign

Le Web est riche sur ce sujet. La somme de travail sur certains sites est stupéfiante. La documentation est d'abord basée sur les pages du W3C traduites par J. J. Solari et mises en ligne sur `www.yoyodesign.org/doc/w3c/css2/cover.html`.

Openweb

Un certain nombre de sites sont incontournables par le travail effectué autour des CSS comme `http://openweb.eu.org/css`. Ce site est remarquable, autant pour son interface que pour ses articles sur les technologies web.

La communication est bien pensée pour satisfaire tout visiteur.

▲ Figure 10.1 : *Communication d'Openweb*

Openweb est tenu par un aréopage de spécialistes coordonnés par Tristan Nitot.

Pompage

Le site Pompage Magazine (`http://pompage.net`) est résolument tourné vers les CSS. Il ouvre au public des textes importants traduits de l'anglais. Il est beau et élégant.

Alsacreations

Le site Alsacreations (`www.alsacreations.fr`), de Raphaël Goetter, est incontournable, aussi bien pour ses tutoriaux sur les CSS et les normes du

W3C, que pour son forum, très actif. Il aborde aussi les thèmes suivants : accessibilité, PHP, JavaScript, etc.

Mediabox

Mediabox (`http://wiki.media-box.net/documentation/css`) est un autre site qui propose un tutorial bien fait sur les CSS.

CSS Zen Garden

Il existe une traduction française du site CSS Zen Garden qui est le must en matière d'expérimentation CSS (`www.csszengarden.com/tr/francais`).

SelfHTML

`http://fr.selfhtml.org/css` est un autre site très complet. C'est une traduction réalisée par Serge François du site original, qui est en allemand. SelfHTML explique en détail les CSS et aussi les autres technologies du Web.

10.2 XHTML

La grange

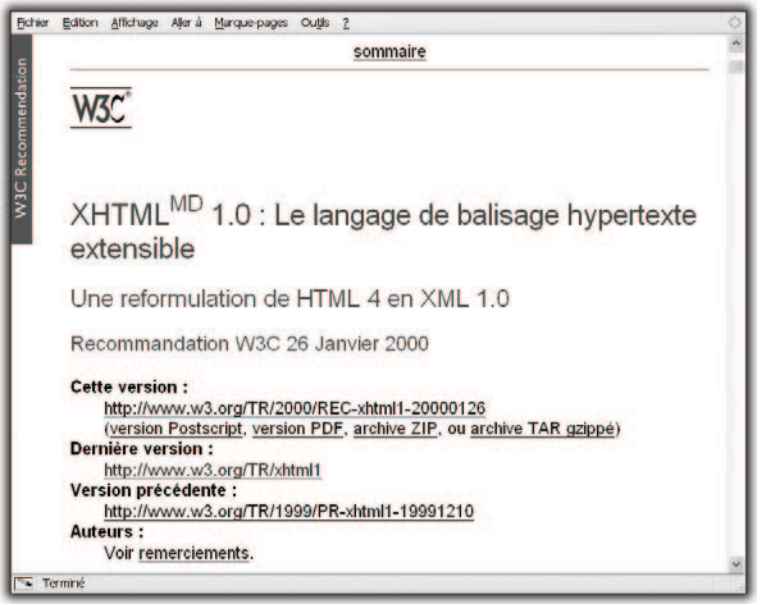
Le site phare est `www.la-grange.net/w3c/xhtml1/`. Il propose la traduction réalisée par Karl Dubost des recommandations du W3C. La plupart des sites déjà mentionnés abordent ce sujet car le XHTML et le CSS sont intimement liés.

All HTML

Le site All HTML (`www.allhtml.com/xhtml/`) est très complet puisqu'il aborde toutes les technologies et développe deux tutoriaux sur ce sujet.

Fusa2

`http://fusa2.free.fr/Webmaster/xhtml.htm` offre de bonnes pages sur le XHTML.



▲ Figure 10.2 : *XHTML*

L'Altruiste

Le site L'Altruiste (`http://www.laltruiste.com`), très complet, aborde toutes les technologies et propose un panorama du XHTML.

10.3 Accessibilité

Accessiweb

L'accessibilité est abordée par la plupart des sites déjà cités. Le site phare est Accessiweb (`www.accessiweb.org`). Il très complet sur le sujet.

La-grange.net

Ce site (`www.la-grange.net/accessibilite/`), très complet et déjà mentionné, propose de faire le tour de l'accessibilité en trente jours.

11

Index

A-B-C

Abbr 661, 668
Absolu 627, 639, 700
Absolute 619, 621
Accessibilité 654, 670
Accesskey 594, 671
Acronym 601, 668
ActiveBorder, ActiveCaption 509
Affectation 723
Aide 674
Alerte 741
Alignement 496, 663
Alinéa 496
Alpha 556
Alt 582, 668
Apostrophe 772
AppWorkspace 509
Arborescence 463
Argument 722, 731, 744
Arrière-plan 510
Ascenseur 538, 588, 595
Attribut 782
Back-office 682
Background 505, 509-512, 709
Balance des blancs 504
Bande passante 538
Bannière 470
Barre de navigation 464
Baseline 496
Bloc 618
Boîte 575, 579, 585, 597, 618, 620, 704
Boîte flottante 553, 597
Booléen 774-775
Bottom 497
Boucle 726
Br 788
Braille 557
Break 729, 743
ButtonFace, ButtonHighlight, ButtonText . 509
Caption, CaptionText 509, 661
Carte 462
Cartouche 609
Case 743
CDATA 785
CFT 502
Chaîne de caractères 776
Chapeau 548, 552
Charset 668
Charte graphique 452
Chroma 557
Classe 529
Clear 597, 634
CMJN 503
Code décimal / hexadécimal 506
Cohérence 514
Coin arrondi 642, 699, 706
Colgroup 663
Color 505, 546
Color Schemer 467, 505
ColorZilla, Colourificator 466
Commentaire 562, 736, 741
Communication 474, 479
Compact 525
Compteur 751
Conforme 608
Constructeur 733
Content 554, 562
Contenu sémantique 463
Continue 729
Cookie 745, 750, 752
Couleurs 439, 470, 502, 540
Critère de nommage 773
CSS, CSS3 ..... 701, 788

D-E-F

Dashed 546
Date 732, 745
Décimal 492, 508
Déclaration 525, 537, 723
Defer 525
Design 452, 455
Disabled 525
Display 500, 619, 763
Display_model 704
Div 471
DOCTYPE 525, 656
Document.cookie 752
Documentation 471
DOM 532
Domain 746
Dpi 494
DTD 525, 782, 785
E-mail 719
EasyPHP 769
Échappement 562, 748
EclipsePalette 467
Em 492, 494
Émulateur 477
Encapsulation 745, 751
Entier 774
Espace de nommage 781, 787
Événement 716, 731
Expires 746
FAQ 454, 554, 674
Favicon.ico 514
Feuille de style 471
Fieldset 659

Filtre 555, 606
First-child 548
First-letter 553
Fixed 595, 619, 625
Float 632, 774-775
Flottant 500, 639, 700
Flux 471, 500
Fonction 722, 731
Font-family, Font-size, etc. 491, 709
For 660, 730
Formulaire 542, 659
Function 731

G-H

Générateurs de métatags 483
GetElementById, GetElementBy etc. 758
GetMonth(),GetTime() 726, 747
Glissière 539
Goat 709
Habillage 538
Handheld 558
Headers 661
Héritage 474, 531
Hiérarchie 513
Highlight, HighlightText 509
History.go() 719
Hr 567, 599
HSL, HSLA 707-708

I-J-L

IconArt 468
Id 660
Identifiant 472
Identification de la page 513
Identité graphique 514
If 726, 743
Illustration 498
Image de fond 615
@import 558-559
Imprimer 718
In 492
InactiveBorder, InactiveCaption 509
Incrémentation 728
IndexOf() 749
InfoBackground, InfoText 509
Inset 546
Instance 732
Int, Integer 774
Ismap 525
Java, JavaScript, JScript 605, 712, 734
Label 660
Lang 536-537, 657
Langue 479, 535
Lettrine 553
Lien 438, 479, 485, 513, 517, 534, 658
Liste 529, 546, 664

M-N-O

Marge externe / interne 565, 630, 638
Master Styler 465
Match 788
Math 738
Média 557
Menu 545, 575, 585
MenuText 509
Meta 481, 668
Min-height, Min-width 563, 638
Mise en page 436, 488
Mmenu 509
Modèle 788
Monospace 490
Moteur de recherche 479, 518
Motif 560
Mots-clés 463, 481-483, 494, 506
Navigation 513, 669
New() 732
Nom 733, 745
Nombre à virgule flottante 775
Non-motif 560
Noresize, Noshade, Nowrap 512, 525
Null 775
Objet 731
OnClick 741
OnSubmit 748
Opacité 556
Opérateur ternaire 743-744
Opquast 590-591
Ordre de lecture 673
Outset 546
Overflow 566, 623

P-R-S

Padding 564-565, 599, 613, 630
Paragraphe 495
ParseInt() 739
Path 746
PCDATA 785
Petite capitale 553
Pica, pixel 492, 494
Positionnement 481
Pourcentage 494, 507
Print 558-559
Progressive rendering 709
Prompt() 739, 749
Pseudo-classe 549-550
Public 461
Puce 666
Random() 738
Règles typographiques 489
Repeat, Repeat-x, Repeat-y 512
Résolution 440, 737
Résumé 463
Return 731
RGB, RGBA 502, 708
Robot 484, 679, 719
Screen 558, 731
Scrollbar 509, 539
Secure 746
Select 545
Sélecteur 526, 534, 562
SetAttribute() 759
Smartpixel 541
Son 708
Sondage 679
Spam 719
Split() 738
Static 619
String 774, 776
Structure d'une page 445
Sub 496
Submit 542
Super 497
Surcharge 474
Switch 743

T-U-V

Table 577, 663
Tableau 661, 724
Taille absolue/relative 491
Tbody 663
Température de couleur 504
Template 702, 788
Temps GMT 747
Text-bottom, Text-indent, Text-top ... 496-497
Tfoot, Th, Thead 661, 663
Thème 455
This 733
ThreeDDarkshadow, ThreeDFace, etc. 509
Title 479, 668
Touche d'accès 671
Transparency 708
TSL 707
TTY 558
Tv 558
Type 723
Typographie 455, 579
Unescape() 748, 754
Validation 475
Variable 723, 773

W-X-Z

WCAG 590
Web Developer 469, 570
Window, WindowFrame, WindowText 509
Window.prompt() 738
XHTML ... 471, 520, 526, 586, 590, 601, 788
XML 658, 779
XSLT 779, 787
Z-index .....

Composé en France par Jouve
11, bd de Sébastopol - 75001 Paris